

本章讨论有限长离散序列的正交变换,主要关注离散傅里叶变换(DFT),详细讨论DFT的定义和性质以及DFT的快速算法FFT。

DFT及其快速算法FFT是数字信号处理中最重要的工具之一,应用非常广泛。本章集中在DFT的定义和性质、DFT的快速算法FFT。有关DFT的各项应用将在后续章节陆续介绍,第4章集中讨论以DFT为核心的数字频谱分析技术,第5章包括了DFT在滤波器实现中的应用,第6章介绍DFT在FIR滤波器设计中的应用,第8章给出了DFT在滤波器组的应用等。

本章也简要介绍了其他的正交变换,例如离散余弦变换(DCT)等。

3.1 离散正交变换

假设 $x[n]$ 只在 $0 \leq n < N$ 区间内取非零值,该区间之外为零,这样的序列称为 N 点有限长序列。对于 N 点有限长序列,最多只有 N 个非零值,故只有 N 个自由度,可以在 N 维矢量空间里完整地表示该信号。在 N 维矢量空间里,定义 N 个相互正交的基序列(相当于基矢量),对于 $0 \leq k \leq N-1$,每个基序列为

$$a_k[n], \quad n = 0, 1, \dots, N-1 \quad (3.1)$$

每个基序列也只定义在区间 $0 \leq n < N$ 内。为了由基序列对有限长序列定义正交变换,要求基序列满足正交性和完备性。

正交性

$$\sum_{n=0}^{N-1} a_k[n] a_{k'}^*[n] = C \delta[k - k'] \quad (3.2)$$

完备性

$$\sum_{k=0}^{N-1} a_k[n] a_k^*[n'] = C \delta[n - n'] \quad (3.3)$$

C 是一个常数,当 $C=1$ 时正交基是归一化的。

由于基序列是正交的,对每个基序列定义一个变换系数 $T[k]$ 为

$$T[k] = \sum_{n=0}^{N-1} x[n] a_k[n], \quad 0 \leq k \leq N-1 \quad (3.4)$$

由完备性,可以证明由全部变换系数 $T[k]$ 可以重构有限长信号为

$$x[n] = \frac{1}{C} \sum_{k=0}^{N-1} T[k] a_k^* [n], \quad 0 \leq n \leq N-1 \quad (3.5)$$

为得到式(3.5),对式(3.4)两侧同乘 $a_k^* [m]$,并对 k 从 $0 \sim N-1$ 求和,得到

$$\begin{aligned} \sum_{k=0}^{N-1} T[k] a_k^* [m] &= \sum_{k=0}^{N-1} a_k^* [m] \sum_{n=0}^{N-1} a_k [n] x[n] = \sum_{n=0}^{N-1} x[n] \sum_{k=0}^{N-1} a_k [n] a_k^* [m] \\ &= C \sum_{n=0}^{N-1} x[n] \delta[n-m] = Cx[m] \end{aligned}$$

上式中将 m 换为 n ,得到式(3.5)。

如前所述,由式(3.4)给出了变换的定义式,然后由完备性导出信号重构公式(3.5)(反变换公式)。容易证明若从式(3.5)出发,两边同乘 $a_k [n]$ 对 n 求和,利用正交性式(3.2),可得到式(3.4),因此,式(3.4)和式(3.5)构成正、反变换对。

满足条件式(3.2)和式(3.3)的基序列有很多。给出每一种基序列,用式(3.4)和式(3.5)可定义一种离散变换,但不是每一种基序列和每一种离散变换都能得到广泛应用。其中有几种离散变换已得到广泛应用,应用最广泛的当属离散傅里叶变换(DFT)。对于按如下定义的基序列

$$a_k [n] = e^{-j\left(\frac{2\pi}{N}k\right)n}, \quad k=0,1,\dots,N-1; n=0,1,\dots,N-1$$

其正交性表示为

$$\sum_{n=0}^{N-1} e^{-j\frac{2\pi}{N}kn} (e^{-j\frac{2\pi}{N}rn})^* = N\delta[k-r]$$

完备性表示为

$$\sum_{k=0}^{N-1} e^{j\frac{2\pi}{N}kn} (e^{j\frac{2\pi}{N}kl})^* = N\delta[n-l]$$

用 $X[k]$ 表示变换系数的 DFT 变换对定义为

$$X[k] = \text{DFT}\{x[n]\} = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}, \quad k=0,1,\dots,N-1 \quad (3.6)$$

$$x[n] = \text{IDFT}\{X[k]\} = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}kn}, \quad n=0,1,\dots,N-1 \quad (3.7)$$

这是利用有限长序列正交展开引出的 DFT 变换的定义。由于 DFT 应用十分广泛,为了理解 DFT 定义的多方面性质,3.2 节我们从其他不同方面重新引入 DFT 的定义。当然这些来自于不同方面对 DFT 的定义式是相同的,但却帮助读者从不同侧面理解 DFT。

3.2 离散傅里叶变换

在 2.4.3 节讨论周期序列的傅里叶变换时我们已经看到,周期序列的傅里叶变换是线谱形式,每个冲激的幅度由对单周期信号的 DTFT 在频点 $\omega_k = \frac{2\pi}{N}k$ 的采样值确定。由此可以得到结论,只需单周期信号 DTFT 在频点 $\omega_k = \frac{2\pi}{N}k$ 的采样值就可以准确表示一个周期信号。那么很自然,它也能够准确表示这个有限长的单周期信号。因此,对于有限长信号,我

们并不需要 DTFT 的全部,而是只需要对 DTFT 在一些离散频点的采样值。

假设 $x[n]$ 是有限长信号,只有在 $0 \leq n < N$ 区间内取非零值,这样,只在 $0 \leq n < N$ 区间内使用式(2.59),并结合式(2.61)得到如下的变换对

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}, \quad k = 0, 1, \dots, N-1 \quad (3.8)$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}kn}, \quad n = 0, 1, \dots, N-1 \quad (3.9)$$

我们称这一变换对为离散傅里叶变换对(DFT),其中式(3.8)是离散傅里叶变换对的正变换(DFT),式(3.9)是离散傅里叶变换对的反变换(IDFT)。

当 $x[n]$ 是有限长信号,只有在 $0 \leq n < N$ 区间内取非零值,此区间之外取零值时,其 DFT 和 DTFT 关系为

$$X[k] = X(e^{j\omega}) \Big|_{\omega = \frac{2\pi}{N}k} = \sum_{n=0}^{N-1} x[n] e^{-j\omega n} \Big|_{\omega = \frac{2\pi}{N}k} \quad (3.10)$$

由式(3.10),我们可以看到,DFT 变换系数 $X[k]$ 有清晰的物理意义。

3.2.1 DFT 作为对 DTFT 的频域离散采样

为了进一步理清一个离散信号的 DFT 和 DTFT 之间的关系、明确 DFT 的物理意义和限制条件,以下我们从对一般离散信号的 DTFT 采样入手,重新审视 DFT 定义的含义和限制。

从一般信号 $x[n]$ 出发,讨论对 $x[n]$ 的 DTFT $X(e^{j\omega})$ 采样问题。这里先不限制 $x[n]$ 为有限长序列,在频点 $\omega_k = \frac{2\pi}{N}k$ 上对 $X(e^{j\omega})$ 进行采样,在 2π 的一个周期内可以采得 N 个点,其他的采样值都是周期重复的。采样值写为

$$X[k] = X(e^{j\omega}) \Big|_{\omega_k = \frac{2\pi}{N}k} = \sum_{n=-\infty}^{+\infty} x[n] e^{-j\frac{2\pi}{N}kn} \quad (3.11)$$

为了使用 DTFT 的反变换公式且仅用 DTFT 的采样值求取原信号,我们定义采样离散时间傅里叶变换为

$$\hat{X}(e^{j\omega}) = \sum_{k=-\infty}^{+\infty} \frac{2\pi}{N} X[k] \delta\left(\omega - \frac{2\pi}{N}k\right) \quad (3.12)$$

将 $\hat{X}(e^{j\omega})$ 代入 DTFT 反变换公式,仅利用 DTFT 的采样值得到的时域离散信号用 $\hat{x}[n]$ 表示,有

$$\begin{aligned} \hat{x}[n] &= \frac{1}{2\pi} \int_0^{2\pi} \hat{X}(e^{j\omega}) e^{j\omega n} d\omega \\ &= \frac{1}{2\pi} \int_0^{2\pi} \sum_{k=-\infty}^{+\infty} \frac{2\pi}{N} X[k] \delta\left(\omega - \frac{2\pi}{N}k\right) e^{j\omega n} d\omega \\ &= \frac{1}{N} \sum_{k=0}^{N-1} X[k] \int_0^{2\pi} \delta\left(\omega - \frac{2\pi}{N}k\right) e^{j\omega n} d\omega \\ &= \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}kn} \end{aligned} \quad (3.13)$$

根据冲激函数的抽样性质,式(3.12)的 $\hat{X}(e^{j\omega})$ 也可以写成

$$\hat{X}(e^{j\omega}) = X(e^{j\omega}) \frac{2\pi}{N} \sum_{k=-\infty}^{+\infty} \delta\left(\omega - \frac{2\pi}{N}k\right) \quad (3.14)$$

由卷积定理和周期冲激串的傅里叶变换得

$$\hat{x}[n] = x[n] * \sum_{r=-\infty}^{+\infty} \delta[n - rN] = \sum_{r=-\infty}^{+\infty} x[n - rN] \quad (3.15)$$

从式(3.15)可以看出,对于无穷长序列 $x[n]$,仅由 DTFT 的采样值 $X[k]$ 是无法得到原信号的,得到的是原信号的重叠相加后的一个周期化序列,这种现象称为时域混叠。只有当 $x[n]$ 是一个有限长序列,其不为零的值集中在 $0 \leq n < M$ 且 $M \leq N$ 时,式(3.15)中各 $x[n - rN]$ 项对不同的 r 互不重叠,此时没有时域混叠发生。只要取出 $\hat{x}(n)$ 中 $0 \leq n < N$ 对应的一个周期的值,就可得到 $x[n]$,即

$$x[n] = \hat{x}[n]R_N[n]$$

总结如上的讨论:当 $x[n]$ 是长度不超过 N 的有限长序列,由对 DTFT 的采样所获得的 $X[k]$ 可完全恢复信号 $x[n]$ 。此时变换对式(3.8)和式(3.9)成立,称为 DFT 变换对。其中 $X[k]$ 是对 $X(e^{j\omega})$ 的采样值,由式(3.10)确定。当 $x[n]$ 是长于 N 的序列时,由采样值 $X[k]$ 所恢复的信号是如式(3.15)所示的混叠后的周期化序列,周期是 N 。

DFT 变换有几个明显的特点。其一,它是可直接计算的。不像 DTFT,由于 ω 取值的连续性,无法编写一个程序或通过一个数字系统直接计算出所有 DTFT 的值。当 $x[n]$ 是 N 点有限长序列,直接通过式(3.8)计算出所有变换系数 $X[k](k=0,1,\dots,N-1)$ 需要 N^2 次复数乘法和 $N(N-1)$ 次复数加法。进一步的研究表明,DFT 存在高效算法,这些高效算法统称快速傅里叶变换(FFT),可以把计算量降低到 $N \log_2 N$ 的量级。

其二,DFT 有明确的物理意义。它是对 DTFT 的采样,表示在频点 $\omega_k = \frac{2\pi}{N}k$ 处的频谱密度值。实际频谱分析的软件或仪器,大多采用 DFT 来计算频谱密度,再通过以 $X[k]$ 为基点插值得到频谱密度 $X(e^{j\omega})$ 。在这些软件或仪器中,是通过用 $X[k]$ 插值来显示频谱图 $X(e^{j\omega})$ 的。容易证明,对于 N 点序列 $x[n]$,由 $X[k]$ 通过插值可获得 $X(e^{j\omega})$ 精确重构。插值公式为

$$X(e^{j\omega}) = \sum_{k=0}^{N-1} X[k] \varphi\left(\omega - \frac{2\pi}{N}k\right) \quad (3.16)$$

其中

$$\varphi(\omega) = \frac{1}{N} e^{-j\frac{N-1}{2}\omega} \frac{\sin \frac{N}{2}\omega}{\sin \frac{\omega}{2}}$$

式(3.16)推导如下

$$\begin{aligned} X(e^{j\omega}) &= \sum_{n=0}^{N-1} x[n] e^{-j\omega n} = \sum_{n=0}^{N-1} \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}kn} e^{-j\omega n} \\ &= \sum_{k=0}^{N-1} X[k] \sum_{n=0}^{N-1} \frac{1}{N} e^{j\frac{2\pi}{N}kn} e^{-j\omega n} = \sum_{k=0}^{N-1} X[k] \frac{1}{N} \frac{1 - e^{-jN(\omega - \frac{2\pi}{N}k)}}{1 - e^{-j(\omega - \frac{2\pi}{N}k)}} \end{aligned}$$

$$= \sum_{k=0}^{N-1} X[k] \frac{1}{N} e^{-j\frac{N-1}{2}(\omega - \frac{2\pi}{N}k)} \frac{\sin \left[\frac{N}{2} \left(\omega - \frac{2\pi}{N}k \right) \right]}{\sin \left[\frac{1}{2} \left(\omega - \frac{2\pi}{N}k \right) \right]}$$

由 $X[k]$ 通过式 (3.16) 可精确重构 $X(e^{j\omega})$, 但在实际应用中, 例如频谱分析的软件包或频谱分析仪中, 显示 $X(e^{j\omega})$ 的幅度谱和相位谱的显示器分辨率是有限的, 并不需要完全精确的 $X(e^{j\omega})$ 值, 可以用一些更简单的插值方式由 $X[k]$ 得到 $X(e^{j\omega})$ 的近似值。例如, 在 N 充分大时, 可以用直线连接相邻 $X[k]$ 的值, 即用线性插值代替式 (3.16) 的精确插值 (见图 3.1), 或者将相邻几个 $X[k]$ 的值用多项式插值来近似 $X(e^{j\omega})$ 。

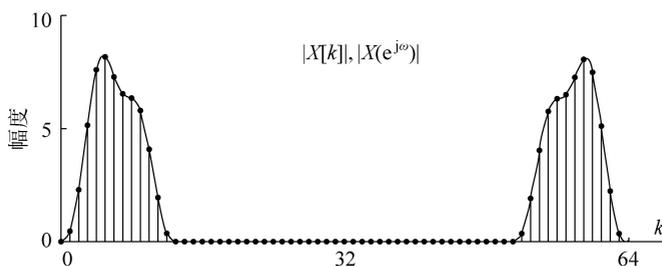


图 3.1 序列的 DFT (垂直点线) 和插值得到的 DTFT (包络线) (图中只画出幅度谱)

其三, 可以利用 DFT 来计算离散 LTI 系统的输出。在一定条件下, 利用 DFT 的快速算法 FFT, 可获得更高效率的卷积计算方法。对该问题, 在 3.4 节介绍 DFT 性质时再详细讨论, 其在滤波器实现中的应用在第 5 章再做进一步分析。

3.2.2 DFT 的矩阵表示

DFT 和 IDFT 可以用矩阵运算形式表示, 我们定义信号矢量

$$\mathbf{x} = [x[0], x[1], \dots, x[N-1]]^T$$

变换系数矢量

$$\mathbf{X} = [X[0], X[1], \dots, X[N-1]]^T$$

变换矩阵

$$\mathbf{T} = [e^{-j\frac{2\pi}{N}kn}]_{N \times N} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & e^{-j\frac{2\pi}{N}} & \cdots & e^{-j\frac{2\pi}{N}(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-j\frac{2\pi}{N}(N-1)} & \cdots & e^{-j\frac{2\pi}{N}(N-1)(N-1)} \end{bmatrix}$$

反变换矩阵

$$\mathbf{U} = \frac{1}{N} [e^{j\frac{2\pi}{N}nk}]_{N \times N} = \frac{1}{N} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & e^{j\frac{2\pi}{N}} & \cdots & e^{j\frac{2\pi}{N}(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{j\frac{2\pi}{N}(N-1)} & \cdots & e^{j\frac{2\pi}{N}(N-1)(N-1)} \end{bmatrix}$$

DFT 的矩阵形式为

$$\mathbf{X} = \mathbf{T}\mathbf{x} \quad (3.17)$$

IDFT 的矩阵形式为

$$\mathbf{x} = \mathbf{U}\mathbf{X} \quad (3.18)$$

不难验证

$$\mathbf{U}\mathbf{T} = \mathbf{I} \quad (3.19)$$

和

$$\mathbf{U} = \frac{1}{N}\mathbf{T}^H \quad (3.20)$$

式(3.19)表示了变换的完备性。通过式(3.17)的变换获得一组变换系数,通过式(3.18)的反变换重构原信号矢量,在这个过程中,能够通过变换和反变换准确重构原信号的条件就是式(3.19)。这可以通过将式(3.17)代入式(3.18)得到验证

$$\mathbf{x} = \mathbf{U}\mathbf{X} = \mathbf{U}\mathbf{T}\mathbf{x} \quad (3.21)$$

由式(3.21)知,只有满足式(3.19)才能通过变换系数准确重构原信号矢量。

3.2.3 DFT 的实例

通过以下例子,进一步理解 DFT 的定义,尤其是通过例子说明 DFT 定义隐含的灵活性。

例 3.2.1 N 点有限长信号

$$x[n] = R_N[n] = \begin{cases} 1, & 0 \leq n < N \\ 0, & \text{其他} \end{cases}$$

做 N 点 DFT, DFT 系数记为 $X[k]$, 由 DFT 的定义得

$$X[0] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn} \Big|_{k=0} = \sum_{n=0}^{N-1} x[n] = N$$

对于 $k \neq 0, 1 \leq k < N$, 有

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn} = \sum_{n=0}^{N-1} e^{-j\frac{2\pi}{N}kn} = 0$$

其 N 点 DFT 系数除 $X[0]$ 外均为零。由这组 DFT 系数,可以反变换得到原信号,对于 $0 \leq n < N$, 有

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}kn} = \frac{1}{N} X[0] e^{j\frac{2\pi}{N}0n} = 1, \quad 0 \leq n < N$$

从另外一个角度理解 DFT 系数, 2.4 节得到 $x[n]$ 的 DTFT 为

$$X(e^{j\omega}) = e^{-j(N-1)\omega/2} \frac{\sin(N\omega/2)}{\sin(\omega/2)}$$

对 DTFT 在 $\omega = \frac{2\pi}{N}k$ 处采样, 首先考虑 $k=0$ 的情况

$$X[0] = X(e^{j\omega}) \Big|_{\omega=0} = N$$

对于 $k \neq 0, 1 \leq k < N$, 有

$$X[k] = X(e^{j\omega}) \Big|_{\omega=\frac{2\pi}{N}k} = e^{-j(N-1)\frac{2\pi}{N}k} \frac{\sin\left(N\frac{2\pi}{2N}k\right)}{\sin\left(\frac{2\pi}{2N}k\right)} = e^{-j(N-1)\frac{\pi}{N}k} \frac{\sin(\pi k)}{\sin\left(\frac{\pi}{N}k\right)} = 0$$

除了线性相位项 $e^{-j(N-1)\omega/2}$ 外, $X(e^{j\omega})$ 重画于图 3.2 中, 可以看到每个过零点位置对应 $N\omega/2 = k\pi$, 即

$$\omega = \frac{2\pi}{N}k$$

处, 这与 DFT 对 DTFT 的采样点重合, 故除 $X[0]$ 外, 其余 DFT 系数均为 0。

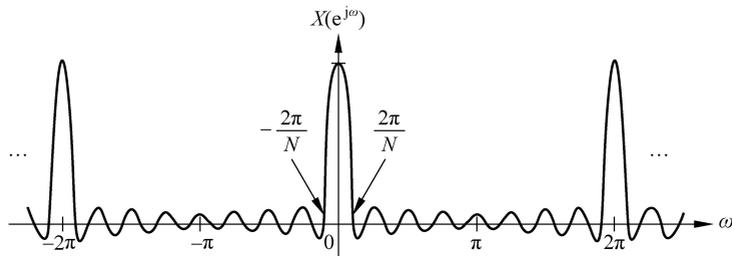


图 3.2 N 点等值序列的 DTFT 示意图(忽略线性相位项)

在例 3.2.1 中, $x[n]$ 有 N 个非零值, 至少要做 N 点 DFT, 才能由变换系数求得原序列的全部非零值。在实际中为了处理的方便, 对于有 N 个非零值的序列, 可以进行 $N_1 > N$ 点的 DFT。也就是说, 在 $x[n]$ 后面补上若干零, 把 $x[n]$ 看成是 N_1 长度的序列进行 DFT。这样, 用 N_1 点 DFT 系数除了可以计算出 $x[n]$ 的全部非零值, 还会把后面补的零值也计算出来, 这在应用中没有什么危害。所以, 对于只有 N 个非零值的序列, 通过补零计算更长点数的 DFT, 是 DFT 应用中经常采用的一种办法。后续还会看到, DFT 的这种灵活性, 大大地拓宽了 DFT 的应用范围。

例 3.2.2 继续讨论例 3.2.1 中的 $x[n]$ 序列, 取 $N_1 = 2N$, 对 $x[n]$ 做 N_1 点 DFT。为了与上例对比, 用 $x_1[n]$ 表示补零后的 $x[n]$, 用 $X_1[k]$ 表示对 $x[n]$ 补零后的 $2N$ 点 DFT 系数。因此

$$x_1[n] = \begin{cases} x[n], & 0 \leq n < N \\ 0, & N \leq n < 2N \end{cases}$$

直接代入 DFT 定义式, 得

$$\begin{aligned} X_1[k] &= \sum_{n=0}^{2N-1} x_1[n] e^{-j\frac{2\pi}{2N}kn} = \sum_{n=0}^{N-1} e^{-j\frac{\pi}{N}kn} \\ &= \begin{cases} N, & k=0 \\ \frac{1 - e^{-j\frac{\pi}{N}kN}}{1 - e^{-j\frac{\pi}{N}k}}, & 1 \leq k < 2N \end{cases} = \begin{cases} N, & k=0 \\ \frac{1 - (-1)^k}{1 - e^{-j\frac{\pi}{N}k}}, & 1 \leq k < 2N \end{cases} \\ &= \begin{cases} N, & k=0 \\ 0, & k \text{ 为偶数} \\ e^{-j(\frac{\pi}{2} - \frac{\pi}{2N}k)} \frac{1}{\sin(\frac{\pi}{2N}k)}, & k \text{ 为奇数} \end{cases} \end{aligned}$$

请读者自行验证, 用

$$X_1[k] = X(e^{j\omega}) \Big|_{\omega = \frac{2\pi}{2N}k}$$

也得到相同的 $X_1[k]$ 表达式。对 $x[n]$ 补上 N 个零, 做 $2N$ 点的 DFT, 相当于对 $X(e^{j\omega})$ 在

$$\omega = \frac{2\pi}{2N}k = \frac{\pi}{N}k$$

处采样, 采样密度比作 N 点 DFT 高了一倍, 相当于对图 3.2 采样时, 将两个过零点之间的峰值处也进行了采样, 这正是 $X_1[k]$ 中 k 为奇数点的那些取值。

补零后做更长点数的 DFT, 一方面, 可得到对 DTFT 更密集(更细致)的采样; 另一方面, 通过反变换仍准确重构信号值, 这给 DFT 带来灵活性。但也注意到, 既然通过 N 点 DFT 就可准确重构信号和插值得到 $X(e^{j\omega})$, 因此, 理论上补零做更长点数 DFT 不会带来更多的有用信息, 但在实际应用中会带来很多方便性。至于 DFT 的这种灵活性带来的方便, 本章后文还会多次遇到。

3.3 DFT 与周期序列傅里叶级数的关系

为讨论方便, 重写离散周期序列的傅里叶级数(DFS)和 DFT 以便于比较。周期序列 $\tilde{x}[n] = \sum_{r=-\infty}^{+\infty} x[n - rN]$, 周期为 N , 其傅里叶级数系数为 $\tilde{X}[k]$ 。尽管只关心 $\tilde{X}[k]$ 的 N 个取值, 但它也是周期为 N 的序列。用简化符号 $W_N = e^{-j\frac{2\pi}{N}}$, 离散周期序列的傅里叶级数展开重写如下

$$\tilde{X}[k] = \sum_{n=0}^{N-1} \tilde{x}[n] W_N^{kn} \quad (3.22)$$

$$\tilde{x}[n] = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}[k] W_N^{-kn} \quad (3.23)$$

考虑有限长序列 $x[n]$, 长度为 N , 序号范围为 $0 \leq n \leq N-1$, 其 DFT 重写为

$$X[k] = \text{DFT}\{x[n]\} = \sum_{n=0}^{N-1} x[n] W_N^{kn}, \quad 0 \leq k \leq N-1 \quad (3.24)$$

$$x[n] = \text{IDFT}\{X[k]\} = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn}, \quad 0 \leq n \leq N-1 \quad (3.25)$$

DFS 和 DFT 仅从数学形式上是一致的, 但两者的目标不同。DFT 的定义中, 只关心有限长序列, 它本质上是对 DTFT 的采样。由此, 其可以看成是时域/频域双域采样得到的“纯离散变换”。尽管 DFT 定义时仅关心有限长序列, 但是因为时域采样带来的频域周期拓展, 使得 DTFT 自身就是周期的, 作为 DTFT 采样的 $X[k]$ 自然具有周期性(尽管不关心 $0 \leq k \leq N-1$ 之外的值); $X[k]$ 作为对 DTFT 的采样, 又引起由 $X[k]$ 重构的信号构成周期拓展, 尽管只关心由式(3.25)在 $0 \leq n \leq N-1$ 内重构 $x[n]$, 但式(3.25)也隐含地满足周期性, 即在 $0 \leq n \leq N-1$ 之外按周期 N 重复 $x[n]$ 的取值。所以, 尽管 DFT 是仅就有限长序列定义的, 但其存在自然的周期延拓性, 即 $X[k]$ 隐含周期性。

为后面讨论方便, 这里给出将有限长序列周期化和从周期序列取出一个周期形成有限长序列的表达方式。由有限长序列 $x[n]$, 得到周期序列 $\tilde{x}[n]$ 可表示为

$$\tilde{x}[n] = \sum_{r=-\infty}^{+\infty} x[n + rN] = x[(n)_N] \quad (3.26)$$

这里 $((\cdot))_N$ 表示对 N 取余数运算,例如 $x[((r+kN))_N]=x[r]$ 。由周期序列,取出一个周期表示为

$$x[n]=\bar{x}[n]R_N[n]=x[((n))_N]R_N[n] \quad (3.27)$$

我们在讨论DFT性质时,尤其是与反转、位移等相关联的性质时,有限长序列位移或反转后序号超出 $0 \leq n \leq N-1$,但由于DFT对 $0 \leq n \leq N-1$ 之外没有定义,不方便于这些性质的理解。为了处理这些性质,可先将有限长序列拓展成周期序列,进行反转或位移后再取其中一个周期。将看到,在表述DFT的一些性质时,就是用的这种方式。

至此,已讨论了各种不同的傅里叶变换。各种傅里叶变换在时域和频域有一些不同的对偶关系,把理想采样信号的傅里叶变换也包括在内,几种傅里叶变换时域和频域的一些对偶关系总结在表3.1中,时域和频域的一般对偶性总结在表3.2中。

表 3.1 各种傅里叶变换时域和频域对偶关系

变换名称	时域性质	频域性质
连续傅里叶级数	周期连续	离散,无限
连续傅里叶变换	连续,无限	连续,无限
采样信号	离散,无限	周期连续
DTFT	离散,无限	周期连续,频域归一化
离散傅里叶级数	离散,周期	离散,周期
DFT	离散,有限长	离散,有限长

表 3.2 时域和频域一般对偶性

时域	频域
连续,无限	连续,无限
离散,无限	周期连续
周期连续	离散,无限
离散,周期	离散,周期

3.4 DFT 的性质

DFT有许多性质,其中一些是简单和自明的,另一些性质却并不直观,需要仔细地解释。对于一些简单性质,只简单列出,证明留作练习。对于几个需要仔细解释的性质,给出详细的讨论。

在叙述DFT性质之前,为了在各性质叙述中节省重复声明的篇幅,假设 $X[k]=\text{DFT}\{x[n]\}$, $X_1[k]=\text{DFT}\{x_1[n]\}$, $X_2[k]=\text{DFT}\{x_2[n]\}$ 。这里 $\text{DFT}\{x[n]\}$ 表示对 $x[n]$ 求离散傅里叶变换,若不加说明,表示 N 点DFT。

性质1 线性性质

若 $x[n]=a_1x_1[n]+a_2x_2[n]$,则 $X[k]=a_1X_1[k]+a_2X_2[k]$ 。线性性质中,各序列均取 N 点DFT,若 $x_1[n]$ 和 $x_2[n]$ 长度不同,则 N 代表较长序列的长度,较短的序列通过补零也做 N 点DFT。

性质2 反转性

对于定义在 $0 \leq n \leq N-1$ 的 N 点有限长序列,如何描述其反转性?信号的反转对应于DFT系数如何变化?对有限长序列的反转性,既要有“反转”之表现,又要保持在 $0 \leq n \leq N-1$ 区间。为了研究反转性,首先将 $x[n]$ 周期化,然后反转,再取其中一个周期值,相当于

$$x[((-n))_N]R_N[n]=\begin{cases} x[0], & n=0 \\ x[N-n], & n \neq 0 \end{cases} \quad (3.28)$$

反转的表现为 $x[0]$ 不变,其他值在取值范围内依次前后对换。为了使表示式更简洁,可将

反转运算简记为

$$x [((-n))_N] R_N [n] = x [N - n] \quad (3.29)$$

例 3.4.1 序列反转的示意图如图 3.3 所示, 图中示出了由 $x[n] = \{5, 4, 3, 2, 1\}$ 周期延拓得到 $x [((n))_N]$, 然后反转得到 $x [((-n))_N]$, 最后仅取第一个周期得到 $x [((-n))_N] R_N [n]$, 最后得到的反转序列如式(3.28)所示。

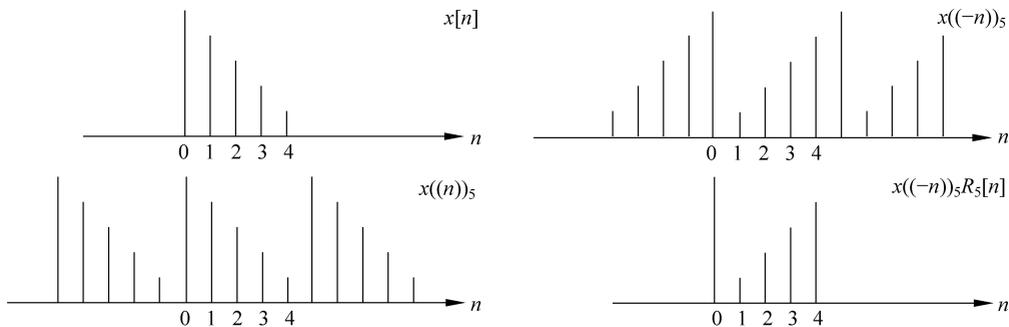


图 3.3 有限长序列反转的示意图

DFT 变换的反转性叙述为

$$\text{DFT} \{x [((-n))_N] R_N [n]\} = X [((-k))_N] R_N [k]$$

或简记为

$$X [N - k] = \text{DFT} \{x [N - n]\} \quad (3.30)$$

即信号反转对应其 DFT 系数反转。

证明

$$\begin{aligned} \text{DFT} \{x [((-n))_N]\} &= \sum_{n=0}^{N-1} x [((-n))_N] e^{-j\frac{2\pi}{N}nk} = x [0] + \sum_{n=1}^{N-1} x [N - n] e^{-j\frac{2\pi}{N}nk} \\ &= x [0] + \sum_{n'=1}^{N-1} x [n'] e^{-j\frac{2\pi}{N}n'(((-k))_N)} = \sum_{n'=0}^{N-1} x [n'] e^{-j\frac{2\pi}{N}n'(((-k))_N)} \\ &= X [((-k))_N] \end{aligned}$$

上式的推导过程中, 由于求和号已经限制了序列的取值范围, 故省略了 $R_N [n]$ 和 $R_N [k]$ 。

对于 N 个非零值的有限长序列, 也可通过补零把序列看作 $N_1 > N$ 长的序列, 若对该补零序列做 N_1 点 DFT, 如上反转性质用 N_1 替代 N 进行描述。

例 3.4.2 在例 3.4.1 中, 若对序列补上 5 个零, 将序列作为 $x[n] = \{5, 4, 3, 2, 1, 0, 0, 0, 0, 0\}$ 的 $N_1 = 2N$ 长序列进行反转, 其反转 $x [((-n))_{10}] R_{10} [n]$ 如图 3.4 所示。

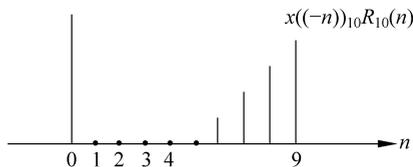


图 3.4 序列补零做反转

性质 3 对偶性(duality)

如果有限长序列 $x[n]$ 的 N 点 DFT 为 $X[k]$, 把 $X[n]$ 看成时域离散序列, 则 $X[n]$ 的 N 点 DFT 为

$$\text{DFT} \{X [n]\} = \sum_{n=0}^{N-1} X [n] e^{-j\frac{2\pi}{N}nk} = \sum_{n=0}^{N-1} \left(\sum_{m=0}^{N-1} x [m] e^{-j\frac{2\pi}{N}mn} \right) e^{-j\frac{2\pi}{N}nk}$$

$$= \sum_{m=0}^{N-1} x[m] \sum_{n=0}^{N-1} e^{-j\frac{2\pi}{N}(m+k)n} = Nx [((-k))_N]$$

可见,如果 $x[n]$ (时域) 和 $X[k]$ (频域) 构成一对离散傅里叶变换对,则 $X[n]$ (时域) 和 $Nx [((-k))_N]$ 之间构成一对离散傅里叶变换对。

利用对偶性,只要知道了某个有限长序列的 DFT,则与其频域序列相同的时域序列的 DFT 也就得到了,无须再重新计算。

性质 4 循环位移性

循环位移性表述为下式的形式

$$\text{DFT}\{x [((n-m))_N] R_N[n]\} = W_N^{km} X[k] \quad (3.31)$$

这里 $x [((n-m))_N] R_N[n]$ 称为循环位移。设 $m > 0$, $x [((n-m))_N]$ 表示周期化后右移 m 点, $R_N[n]$ 取出一个周期。由于 $x [((n-m))_N]$ 的周期性,不难看出, $x [((n-m))_N] R_N[n]$ 相当于 $x[n]$ 依次从 $0 \leq n \leq N-1$ 的尾部移出一个数据,又从头部移进来,直到移动完 m 次。这相当于把 $0 \leq n \leq N-1$ 序列绕在一个圆柱上右移 m 次,所以称这种移位为循环移位。 $m < 0$ 时是相反的移动,数据从头部移出从尾部移入,是循环左移。不管移动方向如何,循环移位对应 DFT 系数乘以 $W_N^{km} = e^{-j\frac{2\pi}{N}km}$ 的指数项。

例 3.4.3 序列 $x[n] = \{5, 4, 3, 2, 1\}$ 按 $N=5$ 进行循环位移,位移量 $m=2$ 的示意图如图 3.5 所示。

上述移位的过程之所以被称为“循环”移位,因为可以形象地用图 3.6 来表示。图 3.6(a) 中有一个圆桶,四周被均匀地划分成 N 个格,依次刻上有限长序列的值。把序列第一点 $x[0]$ 对准 0 刻度压到一根直线轴上,随着圆桶转动,刻在桶上的序列被复制到直线轴上,就形成了周期延拓后的序列。

循环移位,就是先把圆桶旋转指定的步数,正延迟逆时针旋转,负延迟顺时针旋转,然后再把旋转后的圆桶对准 0 刻度压下去,随着圆桶转动,刻在桶上的循环移位后的序列被复制到直线轴上,就形成了延迟后的序列的周期延拓。图 3.6(b) 是延迟为 2 时的示意图。

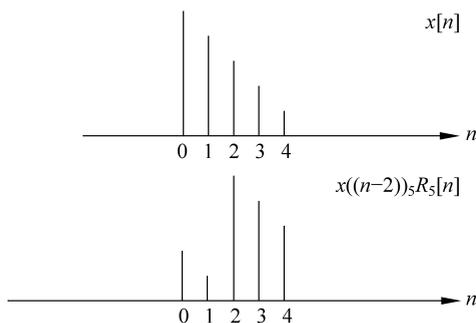


图 3.5 循环位移

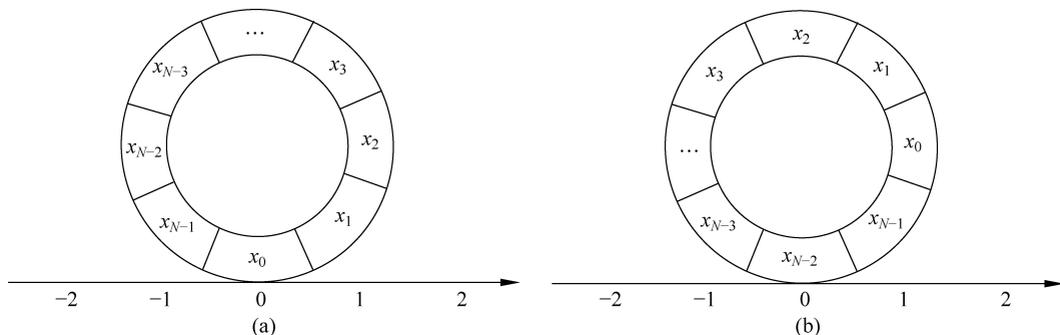


图 3.6 循环移位的圆桶示意图

证明

取 $x_1[n] = x[(n-m)_N]R_N[n]$, 直接代入 DFT 定义运算如下

$$\begin{aligned} \text{DFT}\{x_1[n]\} &= \sum_{n=0}^{N-1} x_1[n] e^{-j\frac{2\pi}{N}nk} = \sum_{n=0}^{N-1} x[(n-m)_N] e^{-j\frac{2\pi}{N}nk} \\ &= \sum_{n=0}^{(m)_N-1} x[n+N-(m)_N] e^{-j\frac{2\pi}{N}nk} + \sum_{n=(m)_N}^{N-1} x[n-(m)_N] e^{-j\frac{2\pi}{N}nk} \\ &= \sum_{n'=N-(m)_N}^{N-1} x[n'] e^{-j\frac{2\pi}{N}n'k} e^{-j\frac{2\pi}{N}(m)_Nk} + \sum_{n'=0}^{N-(m)_N-1} x[n'] e^{-j\frac{2\pi}{N}nk} e^{-j\frac{2\pi}{N}(m)_Nk} \\ &= \sum_{n'=0}^{N-1} x[n'] e^{-j\frac{2\pi}{N}n'k} e^{-j\frac{2\pi}{N}mk} = X[k] e^{-j\frac{2\pi}{N}mk} \end{aligned}$$

这里可能会问, DFT 怎样反映自然移位性(即延迟特性), 即对 N 点有限长序列 $x[n]$, $x[n-m]$ 对应的 DFT 系数怎样? 若 $x[n]$ 在 $0 \leq n \leq N-1$ 范围内的取值均非零, 由于 $x[n-m]$ 的部分数据已移出 $0 \leq n \leq N-1$ 范围, 无法直接用 N 点 DFT 表示, 故无法用 N 点 DFT 性质表述自然移位 $x[n-m]$ 。但是, 这并不是说 DFT 就无法表示自然移位了, 可以通过 DFT 定义的灵活性解决这个问题。

设有一个有限长信号 $x[n]$ 在 $0 \leq n \leq N-1$ 范围内的取值均非零。若应用中需要用到 $m < M$ 的自然移位 $x[n-m]$ 的 DFT 域表示, 这只要把 $x[n]$ 看成是长度为 $N_1 = N+M$ 的序列, 在 $N \leq n < N+M$ 范围 $x[n] = 0$, 对 $x[n]$ 做 N_1 点 DFT。这种做法在 DFT 应用中很常用, 也就是对 $x[n]$ 补 M 个零从而构成长度为 $N_1 = N+M$ 的序列, 再做 N_1 点的 DFT。这样, 当延迟不超过 M 时, $x[n-m]$ 和循环移位 $x[(n-m)_{N_1}]R_{N_1}[n]$ 相等, 注意到, 现在循环移位的循环周期是 N_1 , 这样, $x[n-m]$ 对应的 DFT 系数是 $W_{N_1}^{km} X[k]$, 这里的 $X[k]$ 是 $x[n]$ 补零后的 N_1 点 DFT 系数。

例 3.4.4 将例 3.4.3 中的序列补零, 得到 $x[n] = \{5, 4, 3, 2, 1, 0, 0, 0, 0, 0\}$ 的序列, 按 $N_1 = 10$ 进行循环移位, 得到图 3.7 的位移序列, 这种情况下, 循环位移等于自然位移。

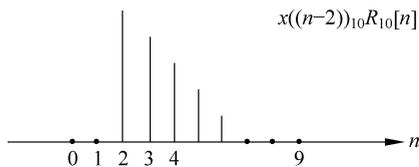


图 3.7 补零后的循环位移

我们将会看到, DFT 的基本性质大多建立在“循环变化”基础上, 但通过补零, DFT 可用于描述各种自然的信号处理算法。

性质 5 频域循环移位性

这是性质 4 的对偶性质, 仅表示如下

$$W_{N_1}^n x[n] \Leftrightarrow X[(k+r)_{N_1}] R_{N_1}[k] \quad (3.32)$$

性质 6 循环卷积性

假设 $x_1[n]$ 和 $x_2[n]$ 均为 N 点序列时, $X_1[k]$ 和 $X_2[k]$ 分别是两个序列的 N 点 DFT。若取 $X[k] = X_1[k]X_2[k]$, 其反变换得到的序列为

$$x[n] = \text{IDFT}\{X[k]\} = \sum_{m=0}^{N-1} x_1[m] x_2[(n-m)_N] \quad (3.33)$$

式(3.33)的求和项称为循环卷积(和)。和式中的第二项是循环反转和位移运算, 尽管只关

心 $0 \leq n \leq N-1$ 范围的结果,但实际上该求和项的结果是周期性的。严格讲,式(3.33)应该写为

$$x[n] = \text{IDFT}\{X[k]\} = \left[\sum_{m=0}^{N-1} x_1[m] x_2[((n-m))_N] \right] R_N(n)$$

由于我们清楚 $x[n]$ 只取 $0 \leq n \leq N-1$ 范围的值,若为简单计可省略 $R_N(n)$ 项。

证明

$$\begin{aligned} x[n] &= \text{IDFT}\{X_1[k]X_2[k]\} = \frac{1}{N} \sum_{k=0}^{N-1} X_1[k]X_2[k]e^{j\frac{2\pi}{N}nk} \\ &= \frac{1}{N} \sum_{k=0}^{N-1} \sum_{m=0}^{N-1} x_1[m]e^{-j\frac{2\pi}{N}mk} X_2[k]e^{j\frac{2\pi}{N}nk} \\ &= \sum_{m=0}^{N-1} x_1[m] \frac{1}{N} \sum_{k=0}^{N-1} X_2[k]e^{-j\frac{2\pi}{N}mk} e^{j\frac{2\pi}{N}nk} \\ &= \sum_{m=0}^{N-1} x_1[m]x_2[((n-m))_N] \end{aligned}$$

注意到上式第3行相当于求 $X_2[k]e^{-j\frac{2\pi}{N}mk}$ 的反变换,故用性质4(循环移位性),得到了第4行的结果。

两个序列 DFT 乘积的 IDFT 是两个序列的循环卷积,循环卷积过程可用图 3.8 形象地表示。图中,两个内外相嵌的圆桶分别被划分成 N 个格,一个序列逆时针排列在其中一个圆桶上,另一个序列顺时针排列在另一个圆桶上。把按照顺时针排列的圆桶按照逆时针方向每次转动一格(或者等价地把按照逆时针排列的圆桶按照顺时针方向每次转动一格),对应格的序列值相乘然后求和,就得到循环卷积结果。

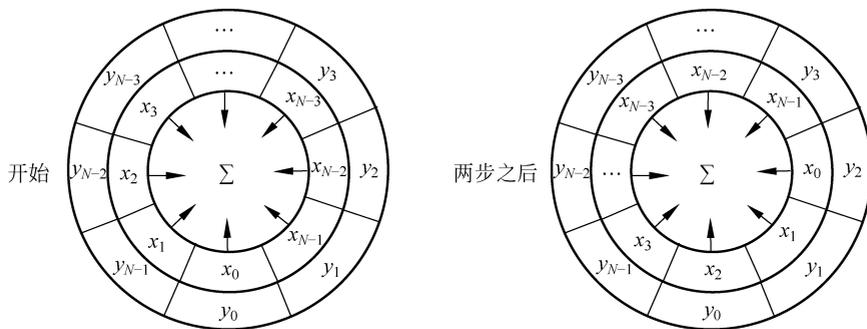


图 3.8 循环卷积的圆桶示意图

循环卷积实际上是两个序列以 N 为周期延拓之后在一个周期上的卷积和。式(3.33)的求和式称为循环卷积,用符号 $x_1[n] \circledast x_2[n]$ 表示。如下讨论循环卷积与线性卷积的关系。为了区分两种卷积,用如下符号表示线性卷积

$$\begin{aligned} x^L[n] &= x_1[n] * x_2[n] = \sum_{m=-\infty}^{+\infty} x_1[m]x_2[n-m] \\ &= \sum_{m=0}^N x_1[m]x_2[n-m] \end{aligned} \quad (3.34)$$

假设 $x_1[n]$ 和 $x_2[n]$ 均为 N 点序列,这是为了使两个序列均可做 N 点 DFT,实际上可以放

宽到 $x_1[n]$ 真正非零长度为 $L \leq N$, $x_2[n]$ 真正非零长度为 $P \leq N$, 通过补零对 $x_1[n]$ 和 $x_2[n]$ 均做 N 点 DFT。由第 2 章介绍的线性卷积性质知, 两个有限长序列 $x_1[n]$ 和 $x_2[n]$ 的线性卷积 $x^L[n]$ 不为零的长度为 $L+P-1$ 。可以证明: 循环卷积与线性卷积的关系为

$$x[n] = \left\{ \sum_{r=-\infty}^{+\infty} x^L[n-rN] \right\} R_N[n] \quad (3.35)$$

这说明, N 点循环卷积是线性卷积以周期为 N 的周期性延拓叠加结果。为了证明式(3.35)的正确性, 对式(3.34)两侧做 DTFT, 得到

$$X^L(e^{j\omega}) = X_1(e^{j\omega})X_2(e^{j\omega}) \quad (3.36)$$

对式(3.36)两侧均在 $\omega_k = \frac{2\pi}{N}k$ 处采样, 得

$$X^L(e^{j\omega}) \Big|_{\omega=\frac{2\pi}{N}k} = X_1(e^{j\omega}) \Big|_{\omega=\frac{2\pi}{N}k} X_2(e^{j\omega}) \Big|_{\omega=\frac{2\pi}{N}k}$$

右侧的采样值分别对应 $X_1[k]$ 和 $X_2[k]$, 故

$$X[k] = X_1[k]X_2[k] = X^L(e^{j\omega}) \Big|_{\omega=\frac{2\pi}{N}k}$$

因此, $X[k]$ 相当于是对 $x^L[n]$ 的 DTFT $X^L(e^{j\omega})$ 在 $\omega_k = \frac{2\pi}{N}k$ 的采样值, 由 DFT 定义时导出的结论式(3.15)得到式(3.35)。

先来看循环卷积和线性卷积最不同的情况。当 $x_1[n]$ 和 $x_2[n]$ 非零长度均为 N , 这时 $x^L[n]$ 是长度为 $2N-1$ 的序列。既然 $X[k]$ 相当于是对 $X^L(e^{j\omega})$ 的 N 点采样 (在 $\omega_k = \frac{2\pi}{N}k$ 的采样值), 由 DFT 定义时的讨论得知, 由 $X[k]$ 无法重构 $2N-1$ 长的序列 $x^L[n]$, 而是得到混叠的结果。这个结果如式(3.35)所示, 由 $X[k]$ 做 IDFT 得到的 $x[n]$ 中几乎所有值都是混叠了的 (只有一点的值是对的)。

上述讨论说明, 无法直接用 DFT 计算线性卷积。但是通过对序列补零, 利用 DFT 的灵活性, 是可以利用 DFT 计算有限长序列的线性卷积的。这里讨论两种特殊情况。

第一种情况, 设 $x_1[n]$ 和 $x_2[n]$ 长度不等, $x_1[n]$ 长度为 L , $x_2[n]$ 长度为 P , 且 $L > P$ 。取 $N=L$, $x_1[n]$ 直接做 N 点 DFT, $x_2[n]$ 补零后也做 N 点 DFT, 对 $X[k]=X_1[k]X_2[k]$ 做 N 点 IDFT 得到 $x[n]$ 。在这些条件下, 通过式(3.35) $x^L[n]$ 和 $x[n]$ 的关系示于图 3.9 中。

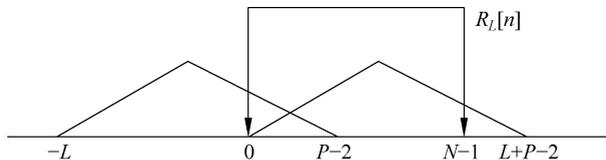


图 3.9 线性卷积和与循环卷积和的关系

如图 3.9 所示, 矩形框住的值就是 IDFT 的结果 $x[n]$, 而三角图形表示线性卷积 $x^L[n]$ 和它的各移位求和项。在该情况下, 式(3.35)中只有 $x^L[n]$ 和 $x^L[n+N]$ 两项对矩形窗内的值有影响, 两个三角形重叠的区域 $x[n]$ 的取值由 $x^L[n]$ 和 $x^L[n+N]$ 混合而成。观察图 3.9 发现, $x[n]$ 在 $0 \leq n \leq P-2$ 范围内的 $P-1$ 个值是混叠结果, 而 $x[n]$ 在 $P-1 \leq n \leq N-1$ 范围内的 $L-P+1$ 个值是与线性卷积相等的。这种情况说明, 当 $x_1[n]$ 和 $x_2[n]$ 长度不等时, 若以长序列的点数做 DFT, 利用 IDFT 计算得到的循环卷积中保留了线性卷积的

部分正确结果,即 $P-1 \leq n \leq N-1$ 范围内循环卷积保留了线性卷积的部分正确值。这个结论在 FIR 滤波器计算时会得到应用。

第二种情况,仍设 $x_1[n]$ 长度为 L , $x_2[n]$ 长度为 P 。取 $N \geq L+P-1$,对 $x_1[n]$ 和 $x_2[n]$ 均补零,做 N 点 DFT,计算 $X[k]=X_1[k]X_2[k]$ 的 IDFT,得到循环卷积 $x[n]$ 。在这种条件下,因满足

$$x[n] = \left\{ \sum_{r=-\infty}^{+\infty} x^L[n-rN] \right\} R_N[n] = x^L[n]$$

循环卷积等于线性卷积。由此,对于有限长序列,用循环卷积计算线性卷积的步骤为取 $N \geq L+P-1$,计算如下:

- (1) $x_1[n]$ 在 $P \sim N-1$ 补零,做 N 点 DFT;
- (2) $x_2[n]$ 在 $L \sim N-1$ 补零,做 N 点 DFT;
- (3) $X[k]=X_1[k]X_2[k]$, $0 \leq k \leq N-1$;
- (4) $x^L[n]=x[n]=\text{IDFT}\{X[k]\}$, $0 \leq n \leq N-1$ 。

用循环卷积计算线性卷积有其实际意义。若把上述讨论中的 $x_1[n]$ 换作一个离散 LTI 系统的单位抽样响应,把 $x_2[n]$ 看作系统的输入信号,则线性卷积就是系统的输出。利用循环卷积也可计算系统的输出。由于 DFT 存在快速算法 FFT,一定条件下用 DFT 计算系统输出可能更经济。

在实际应用中还会遇到 LTI 系统的单位抽样响应是有限长的,即 FIR 系统,但系统输入信号是无限长序列。这种情况下,可将输入序列分成有限长的段进行处理,具体算法将在第 5 章进一步讨论。

例 3.4.5 有两个信号, $x_1[n]=\{1,1,1,1,1\}$, $x_2[n]=\{5,4,3,2,1\}$,取 $N=5$,进行循环卷积,首先利用式(3.33)的定义直接计算循环卷积,计算过程的示意图如图 3.10 所示,记循环卷积的结果为 $y[n]$,显然 $y[n]=\{15,15,15,15,15\}$ 。

图 3.10(c)是 $x_2[((-k))_5]$,注意到

$$y[0] = \sum_{k=0}^4 x_1[k]x_2[((-k))_5] = 15$$

对于 $1 \leq n < 5$,图 3.10(c)进行循环位移,得到 $x_1[k]x_2[((n-k))_5]$ 并进行求和,但由于 $x_1[k]=1$,所以循环卷积结果总是 $x_2[n]$ 非零值之和,恒为 15。

以上结果同样可以通过 DFT 求得。 $N=5$ 时,例 3.2.1 已得到 $x_1[k]$ 的 DFT 系数为:只有 $X_1[0]=5$ 非零,故也只需要求出 $X_2[0]=15$,因此

$$X_1[k]X_2[k] = \{75, 0, 0, 0, 0\}$$

求 IDFT 为

$$y[n] = \frac{1}{5} \sum_{k=0}^4 X_1[k]X_2[k]W_5^{-kn} = \frac{1}{5} \times 75W_5^0 = 15$$

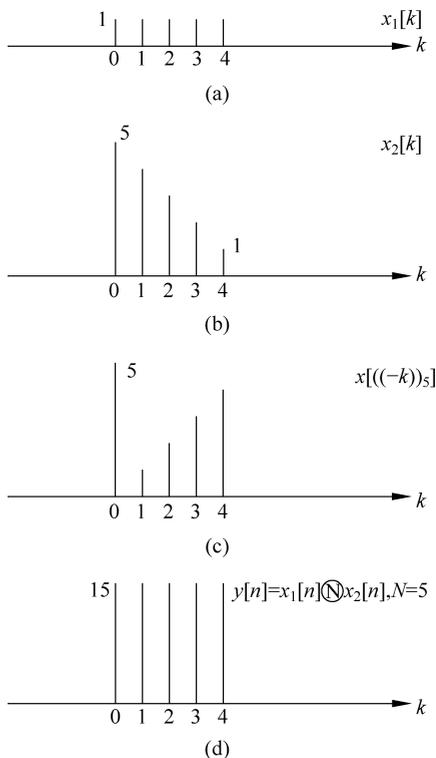


图 3.10 循环卷积

例 3.4.6 离散序列与例 3.4.5 相同,本例取 $N=10$ 进行循环卷积,补零后两个信号为

$$x_1[n] = \{1, 1, 1, 1, 1, 0, 0, 0, 0, 0\}, \quad x_2[n] = \{5, 4, 3, 2, 1, 0, 0, 0, 0, 0\}$$

直接计算循环卷积,示意图如图 3.11 所示。图 3.11(b)和图 3.11(c)给出了 $x_2[(n-k)_{10}]$ 在 $n=0$ 和 $n=2$ 的示意图,图 3.11(d)是循环卷积的结果。可见对本例,取 $N=10$ 时,计算得到的循环卷积等于线性卷积。本例满足循环卷积等于线性卷积的最小 N 值是 9。

同样的结果,可通过 DFT 求得,做 $N=10$ 点 DFT,得

$$X_1[k] = \{5, 1 - j 3.0777, 0, 1 - j 0.7265, 0, 1, 0, 1 + j 0.7265, 0, 1 + j 3.0777\}$$

$$X_2[k] = \{15, 7.7361 - j 7.6942, 2.5 - j 3.4410, 3.2639 - j 1.8164i,$$

$$2.5 - j 0.8123i, 2.5 + j 0.8123, 3.2639 + j 1.8164, 2.5 + j 3.4410,$$

$$7.7361 + j 7.6942\}$$

相乘后做 IDFT 得

$$y[n] = \text{IDFT}\{X_1[k]X_2[k]\} = \{5, 9, 12, 14, 15, 10, 6, 3, 1, 0\}$$

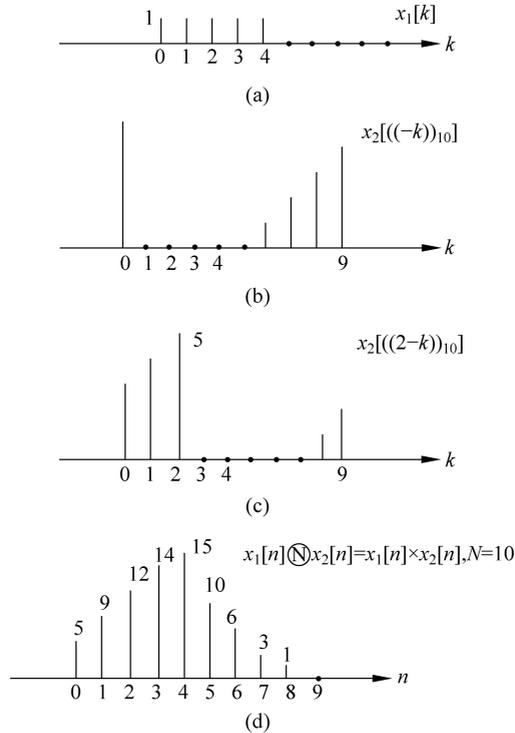


图 3.11 补零后循环卷积等于线性卷积的例子

性质 7 频域卷积性

设序列 $x[n] = x_1[n]x_2[n]$, 其中 $x[n]$ 、 $x_1[n]$ 、 $x_2[n]$ 均为 N 点有限长序列,并分别做 N 点 DFT, 则

$$X[k] = \text{DFT}\{x[n]\} = \sum_{l=0}^{N-1} X_1[l]X_2[(k-l)_N]$$

性质 8 共轭性质

$$\text{DFT}\{x^*[n]\} = X^*[N-k]$$

证明

$$\begin{aligned} \text{DFT}\{x^*[n]\} &= \sum_{n=0}^{N-1} x^*[n] e^{-j\frac{2\pi}{N}kn} = \left(\sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}(-k)n} \right)^* \\ &= \left(\sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}((-k))_N n} \right)^* = \left(\sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}(N-k)n} \right)^* = X^*[N-k] \end{aligned}$$

性质 9 DFT 的对称性

利用有限长序列的周期延拓,定义有限长序列的周期共轭对称性为

$$x_{\text{ep}}[\langle(n)\rangle_N] R_N[n] = x_{\text{ep}}^*[\langle(-n)\rangle_N] R_N[n]$$

记住,除零点自身互共轭(一定是实数)外,周期共轭对称关系可简写为

$$x_{\text{ep}}[n] = x_{\text{ep}}^*[N-n]$$

类似地,周期共轭反对称分量简写为

$$x_{\text{op}}[n] = -x_{\text{op}}^*[N-n]$$

对于一个任意序列 $x[n]$,可分解为周期共轭对称序列和周期共轭反对称序列之和。周期共轭对称序列和周期共轭反对称序列与 $x[n]$ 的关系为

$$x_{\text{ep}}[n] = \frac{1}{2} \{x[n] + x^*[N-n]\}$$

$$x_{\text{op}}[n] = \frac{1}{2} \{x[n] - x^*[N-n]\}$$

如果 $x[n]$ 的 DFT 分别写成其实部和虚部之和,即 $X(k) = X_R(k) + jX_I(k)$,则有

$$X_R[k] = \text{DFT}\{x_{\text{ep}}[n]\}$$

$$jX_I[k] = \text{DFT}\{x_{\text{op}}[n]\}$$

对偶的关系表述为:若 $x[n] = x_r[n] + jx_i[n]$ 为复序列, $x_r[n]$ 是实部, $x_i[n]$ 为虚部,则

$$\text{DFT}\{x_r[n]\} = X_{\text{ep}}[k] = \frac{1}{2} \{X[k] + X^*[N-k]\}$$

$$\text{DFT}\{jx_i[n]\} = X_{\text{op}}[k] = \frac{1}{2} \{X[k] - X^*[N-k]\}$$

例 3.4.7 利用对称性可以简化 DFT 的计算。例如,有两个实值信号 $x_1[n]$ 和 $x_2[n]$,用它们分别作为实部和虚部构成一个复信号 $x[n] = x_1[n] + jx_2[n]$ 。直接计算得到复信号的 DFT $X[k]$,利用对称性得到 $x_1[n]$ 和 $x_2[n]$ 的 DFT 为

$$X_1[k] = \text{DFT}\{x_1[n]\} = X_{\text{ep}}[k] = \frac{1}{2} \{X[k] + X^*[N-k]\}$$

$$X_2[k] = \text{DFT}\{x_2[n]\} = \frac{1}{j} X_{\text{op}}[k] = \frac{1}{2j} \{X[k] - X^*[N-k]\}$$

性质 10 实序列 DFT 的对称性

这是性质 9 的特例,因为重要,故单独列为一条性质。若离散信号是实值信号,其信号自身就是其实部。这说明,实值信号的 DFT 自身满足周期共轭对称性,即

$$X[k] = X^*[N-k] \quad (3.37)$$

也就是说,实值有限长序列的 DFT 系数满足 $X[0]=X^*[0]$,故 $X[0]$ 是实值;其他 DFT 系数在 $1 \leq k \leq N-1$ 范围内,满足式(3.37)的周期共轭对称性质。若 N 是偶数,也可确定 $X[N/2]=X^*[N/2]$ 是实数。

对一个 N 点实序列,其 N 点 DFT 系数存在共轭对称性,仅由 $X[0]$ 、 $X[N/2]$ 和 $1 \leq k \leq N/2-1$ 的系数即可确定全部系数。这样,可以用 $N/2$ 个复数存储单元即可存储 N 点实序列的 DFT 系数。由于存储一个复数需要两个实数存储单元,因此存储实序列和存储其 DFT 系数需要相同数目的存储单元。

性质 11 帕塞瓦尔定理

如果有限长序列 $x[n]$ 的 N 点 DFT 为 $X[k]$,帕塞瓦尔定理(Parseval's Theorem)反映变换前后序列的能量关系

$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X[k]|^2$$

证明

$$\begin{aligned} \sum_{n=0}^{N-1} |x[n]|^2 &= \sum_{n=0}^{N-1} x[n] x^*[n] = \sum_{n=0}^{N-1} \left(\frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}nk} \right) x^*[n] \\ &= \frac{1}{N} \sum_{k=0}^{N-1} X[k] \left(\sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}nk} \right)^* = \frac{1}{N} \sum_{k=0}^{N-1} X[k] X^*[k] \\ &= \frac{1}{N} \sum_{k=0}^{N-1} |X[k]|^2 \end{aligned}$$

DFT 有许多性质,其中一些性质很独特。灵活使用 DFT 的各种性质,可以极大地方便 DFT 的各类应用。

3.5 用 DFT 计算相关序列

如果一有限长离散信号 $x[n]$,在 $0 \leq n \leq N-1$ 内非零,根据自相关的定义,自相关序列为

$$r_{xx}[k] = \sum_{n=-\infty}^{+\infty} x[n] x^*[n-k] = \begin{cases} \sum_{n=k}^{N-1} x[n] x^*[n-k], & 0 \leq k \leq N-1 \\ \sum_{n=0}^{N-1+k} x[n] x^*[n-k], & -N+1 \leq k < 0 \\ 0, & |k| \geq N \end{cases}$$

对于有限长序列来讲,信号总是能量有限的,故用如上的能量信号的自相关定义。自相关取值不为零的区间为 $|k| \leq N-1$,非零长度为 $2N-1$ 。

第 2 章已经详细研究了自相关的 DTFT,假设

$$E_{xx}(e^{j\omega}) = \text{DTFT}\{r_{xx}[k]\} = |X(e^{j\omega})|^2$$

由于自相关是 $2N-1$ 长度的,只要取 $L \geq 2N-1$,在 $\omega_k = \frac{2\pi}{L}k$ 对 $E_{xx}(e^{j\omega}) = |X(e^{j\omega})|^2$ 进行采样,由采样值可重构自相关序列。故得到用 DFT 计算自相关序列的算法如下。

算法 1:

(1) 取 $L \geq 2N - 1$, 离散信号 $x[n]$ 补零, 得到

$$x_L[n] = \begin{cases} x[n], & 0 \leq n \leq N - 1 \\ 0, & N \leq n \leq L - 1 \end{cases}$$

(2) 对 $x_L[n]$ 做 L 点 DFT, 得到 $X_L[k], 0 \leq k \leq L - 1$;

(3) 计算 $E_{xx}[k] = |X_L[k]|^2, 0 \leq k \leq L - 1$;

(4) 计算 IDFT, 得到 $r_L[n] = \text{IDFT}\{|X_L[k]|^2\}, 0 \leq n \leq L - 1$;

(5) 得到自相关序列为

$$r_{xx}[k] = \begin{cases} r_L[k], & 0 \leq k \leq N - 1 \\ r_L[L + k], & -N + 1 \leq k < 0 \\ 0, & |k| \geq N \end{cases}$$

可通过类似的讨论, 给出互相关的计算步骤如下。

算法 2:

(1) 取 $L \geq 2N - 1$, 离散信号 $x[n], y[n]$ 补零, 得到

$$x_L[n] = \begin{cases} x[n], & 0 \leq n \leq N - 1 \\ 0, & N \leq n \leq L - 1 \end{cases}$$

$$y_L[n] = \begin{cases} y[n], & 0 \leq n \leq N - 1 \\ 0, & N \leq n \leq L - 1 \end{cases}$$

(2) 对 $x_L[n]$ 做 L 点 DFT, 得到 $X_L[k], 0 \leq k \leq L - 1$, 对 $y_L[n]$ 做 L 点 DFT, 得到 $Y_L[k], 0 \leq k \leq L - 1$;

(3) 计算 $E_{xy}[k] = X_L[k]Y_L^*[k], 0 \leq k \leq L - 1$;

(4) 计算 IDFT, 得到 $r_{xyL}[n] = \text{IDFT}\{E_{xy}[k]\}, 0 \leq n \leq L - 1$;

(5) 得到互相关序列为

$$r_{xy}[k] = \begin{cases} r_{xyL}[k], & 0 \leq k \leq N - 1 \\ r_{xyL}[L + k], & -N + 1 \leq k < 0 \\ 0, & |k| \geq N \end{cases}$$

在研究了 DFT 的快速算法 FFT 后, 自相关和互相关计算中的 DFT 和 IDFT 都采用 FFT 算法高效实现。

3.6 DFT 的快速计算方法

由 DFT 的定义

$$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{nk} \quad (3.38)$$

计算每一个 DFT 系数需要 N 次复数乘法和 $N - 1$ 次复数加法, 本节不加说明时, 乘法和加法次数均指复数运算。这样计算所有的 DFT 系数需要 N^2 次复数乘法和 $N(N - 1)$ 次复数加法。一次复数乘法需要四次实数乘法和两次实数加法, 一次复数加法需要 2 次实数加法。所

以,直接计算 DFT 所需要的实数乘法次数为 $4N^2$,所需要的实数加法次数为 $N(4N-2)$ 。这样的算法复杂度我们称为 N^2 量级的,简记为 $O(N^2)$ 。在数据采集速率比较高, N 取值较大时,实时计算 DFT 的运算量仍是很可观的。本节研究 DFT 的快速计算方法:快速傅里叶变换(FFT)。

DFT 定义中的 W_N^{nk} 项,有许多性质可用于简化 DFT 的计算。首先,一些特殊值可节省运算,如 $W_N^0 = W_N^{kN} = 1, W_N^{N/2} = -1$,这些项都节省乘法运算;另外一些项如 $W_N^{N/4} = -j, W_N^{3N/4} = j$ 在复数乘法运算时,并不需要实际做乘法,只是交换实部和虚部,可统计为 0 次乘法,这些特殊项均可节省乘法次数。另外一些性质,主要是周期性、齐次性和对称性,与序列分解结合可显著节省运算量。

周期性为

$$W_N^{nk} = W_N^{(N+n)k} = W_N^{n(k+N)}$$

齐次性为

$$W_N^k = W_{N/m}^{k/m}$$

变换基的对称性表现为复共轭对称性(complex conjugate symmetry)

$$W_N^{(N-n)k} = W_N^{-nk} = (W_N^{nk})^*$$

利用变换基的周期性和对称性可以从一定程度上提高计算效率,但是并不是数量级上的改进。提高 DFT 计算效率的另一类方法是把长序列分解成短序列,先做短序列的 DFT,然后再在此基础上进一步得到长序列的 DFT。

应用变换基的这些性质可方便地将长序列分解为短序列进行计算。那些可直接节省乘法运算次数的项,对于长序列 DFT 的直接计算来讲,节省的乘法次数并不明显,但对于短序列可以明显节省乘法次数。例如 $N=2$ 或 $N=4$ 时,可不需要任何乘法运算。为了看清楚这一点,列出如下几个短序列直接做 DFT 的计算公式。

2 点 DFT 的直接计算,不需要乘法,只需要两次加法,计算如下

$$X[k] = \sum_{n=0}^1 x[n]W_2^{nk} = x[0]W_2^0 + x[1]W_2^k, \quad k=0,1$$

计算两个系数的矩阵形式为

$$\begin{bmatrix} X[0] \\ X[1] \end{bmatrix} = \begin{bmatrix} W_2^0 & W_2^0 \\ W_2^0 & W_2^1 \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \end{bmatrix} = \begin{bmatrix} x[0] + x[1] \\ x[0] - x[1] \end{bmatrix}$$

这个计算过程可表示为图 3.12 的蝶形结构。

3 点 DFT 的计算如下

$$X[k] = \sum_{n=0}^2 x[n]W_3^{nk} = x[0]W_3^0 + x[1]W_3^k + x[2]W_3^{2k}, \quad k=0,1,2$$

矩阵形式为

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \end{bmatrix} = \begin{bmatrix} W_3^0 & W_3^0 & W_3^0 \\ W_3^0 & W_3^1 & W_3^2 \\ W_3^0 & W_3^2 & W_3^4 \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & W_3^1 & W_3^2 \\ 1 & W_3^2 & W_3^1 \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \end{bmatrix}$$

3 点 DFT 的蝶形结构示于图 3.13。

4 点 DFT 计算为

$$X[k] = \sum_{n=0}^3 x[n]W_4^{nk} = x[0]W_4^0 + x[1]W_4^k + x[2]W_4^{2k} + x[3]W_4^{3k}, \quad k=0,1,2,3$$

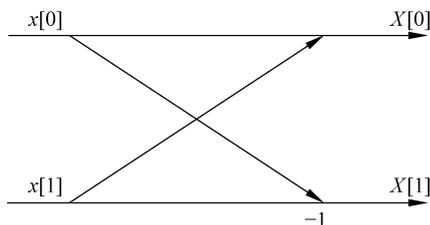


图 3.12 2点 DFT 的蝶形计算

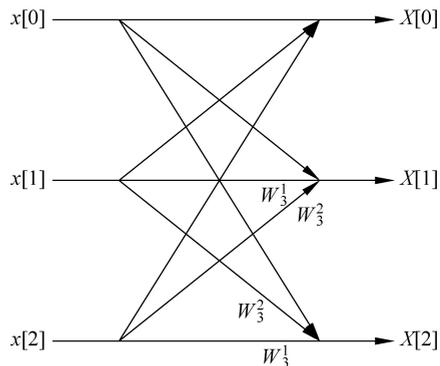


图 3.13 3点 DFT 的蝶形计算

矩阵形式为

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{bmatrix} = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^4 & W_4^6 \\ W_4^0 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix}$$

当 DFT 算法要求的变换点数 N 为组合数时,既然 N 为组合数,比如 $N = N_1 N_2$,那么 n 可以表示为

$$n = n_2 N_1 + n_1, \quad n_1 = 0, 1, \dots, N_1 - 1; \quad n_2 = 0, 1, \dots, N_2 - 1$$

此时

$$W_N^{nk} = W_N^{(n_2 N_1 + n_1)k} = W_N^{n_2 N_1 k} W_N^{n_1 k}$$

而

$$W_N^{n_2 N_1 k} = e^{-j \frac{2\pi}{N} n_2 N_1 k} = e^{-j \frac{2\pi}{N_2} n_2 k} = W_{N_2}^{n_2 k}$$

是 N_2 点 DFT 变换的基。变换基的以上性质对 k 同样适用。

利用这种组合数性质,可以把 N 点长序列的变换基转变成 N_1 或 N_2 点短序列的变换基。经过适当的分解与合成,将对长序列的 DFT 计算问题转换为计算短序列的 DFT 然后再组合,这样做可能大大降低运算量。尤其是短序列因子 N_i 为 2 或 4 时,运算效率更高。这类算法统称为快速傅里叶变换(fast fourier transform, FFT)。注意,FFT 不是一类新变换,而是 DFT 快速计算算法的总称。

快速傅里叶变换可以把 DFT 的计算量减少到 $O(N \lg N)$ 量级。由于计算量小,FFT 得到广泛应用。

但是标准 FFT 算法必须要把所有 $X[k]$ 都计算出来,对于某些只需要一部分谱的应用来说,特别是当点数 N 比较大时,FFT 算法在效率方面并没有优势。此时采用卷积实现 DFT 的线性调频 z 变换(chirp- z)算法可以减少无效计算。

在对序列做滑窗(sliding-window)分析时,如果上次的 FFT 结果已知,则分析窗滑动一步,只有一个最老的值被滑出窗外,同时只有一个新值被滑进窗内,此时可以利用上次 FFT 结果递推得到新的 FFT 结果。这种算法可以把每一段变换的计算量减小到 $O(N)$ 。

3.6.1 按时间抽取基 2-FFT 算法

既然目的是将长序列分解成短序列,先进行短序列的 DFT,将短序列的 DFT 合成为长序列的 DFT,很自然的一种分解方法是将序列分成两部分,偶数序号为一个新的短序列,奇数序号组成第二个短序列,即

$$\begin{cases} e[n] = x[2n], \\ f[n] = x[2n + 1], \end{cases} \quad 0 \leq n \leq \frac{N}{2} - 1 \quad (3.39)$$

这里 $e[n]$ 和 $f[n]$ 都是 $N/2$ 点序列,对其分别 DFT,记为 $E[k]$ 和 $F[k]$,均为 $N/2$ 点 DFT。考虑到利用短序列 DFT 合成长序列 DFT 时,序号 k 在 $0 \leq k \leq N-1$ 范围取值,利用 DFT 隐含的周期性,用 $E[(k)_{N/2}]$ 和 $F[(k)_{N/2}]$ 分别表示 $E[k]$ 和 $F[k]$ 的周期延拓。这样

$$\begin{aligned} X[k] &= \sum_{n=0}^{N-1} x[n] W_N^{nk} \\ &= \sum_{r=0}^{\frac{N}{2}-1} x[2r] W_N^{2rk} + \sum_{r=0}^{\frac{N}{2}-1} x[2r+1] W_N^{(2r+1)k} \\ &= \sum_{r=0}^{\frac{N}{2}-1} e[r] W_{N/2}^{rk} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} f[r] W_{N/2}^{rk} \\ &= E[(k)_{N/2}] + W_N^k F[(k)_{N/2}] \end{aligned} \quad (3.40)$$

上式第三行是 $N/2$ 点 DFT,考虑到 k 在 $0 \leq k \leq N-1$ 范围取值,第四行用了 $N/2$ 点 DFT $E[k]$ 和 $F[k]$ 的周期延拓形式。

式(3.40)给出了用短序列变换合成长序列变换的公式,为了方便,取 $0 \leq k \leq \frac{N}{2} - 1$,重写式(3.40)最后一行为

$$\begin{cases} X[k] = E[k] + W_N^k F[k], \\ X\left[k + \frac{N}{2}\right] = E[k] - W_N^k F[k], \end{cases} \quad 0 \leq k \leq \frac{N}{2} - 1 \quad (3.41)$$

由式(3.41)更清楚地看到,由于 $W_N^k F(k)$ 只需要计算一次,若已经计算出 $N/2$ 点 DFT $E[k]$ 和 $F[k]$,由 $E[k]$ 和 $F[k]$ 得到 $X[k]$, $0 \leq k \leq N-1$,仅需要 N 次加法和 $N/2$ 次乘法。图 3.14 中,以 $N=8$ 为例,画出了由两个 $N/2$ 点 DFT $E[k]$ 和 $F[k]$ 合成得到 $X[k]$ 的示意图。

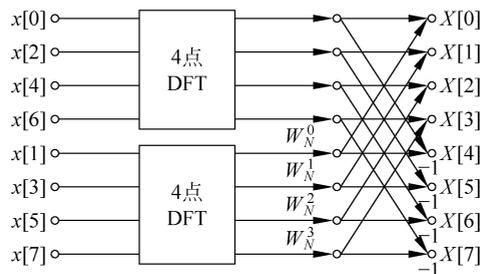


图 3.14 8 点序列第一次分解流程图

图 3.14 中看到,只有奇序列的变换 $F[k]$ 输出端需要乘因子 W_N^k ,该因子称为旋转因子。一次分解后总运算量为

乘法次数

$$2\left(\frac{N}{2}\right)^2 + \frac{N}{2} = \frac{N}{2}(N+1) \approx \frac{N^2}{2}$$

加法次数

$$2\left(\frac{N}{2}\right)\left(\frac{N}{2}-1\right) + N = \frac{N^2}{2}$$

乘法次数里的近似符号对大的 N 成立,可以看到一次分解获得的效果是大约降低了一半的乘法次数和加法次数,因此,这种分解可以继续下去,将 $e[n]$ 和 $f[n]$,按照奇偶继续划分为

$$\begin{cases} a[n] = e[2n] = x[4n], \\ b[n] = e[2n+1] = x[4n+2], \\ c[n] = f[2n] = x[4n+1], \\ d[n] = f[2n+1] = x[4n+3], \end{cases} \quad 0 \leq n \leq \frac{N}{4} - 1$$

对以上序列分别做 $N/4$ 点 DFT,类似式(3.41)的推导,得到

$$\begin{cases} E[k] = A[k] + W_{N/2}^k B[k] = A[k] + W_N^{2k} B[k], \\ E\left[k + \frac{N}{4}\right] = A[k] - W_{N/2}^k B[k] = A[k] - W_N^{2k} B[k], \end{cases} \quad 0 \leq k \leq \frac{N}{4} - 1 \quad (3.42)$$

$$\begin{cases} F[k] = C[k] + W_{N/2}^k D[k] = C[k] + W_N^{2k} D[k], \\ F\left[k + \frac{N}{4}\right] = C[k] - W_{N/2}^k D[k] = C[k] - W_N^{2k} D[k], \end{cases} \quad 0 \leq k \leq \frac{N}{4} - 1 \quad (3.43)$$

以 $N=8$ 为例,图 3.15 画出第二次分解后的示意图。

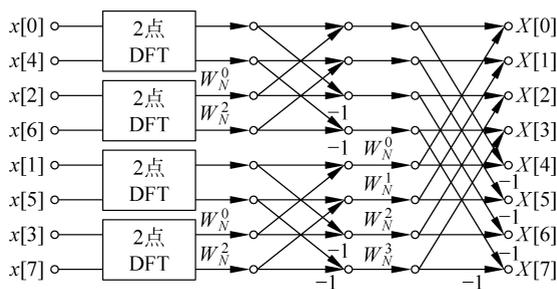


图 3.15 8 点序列第二次分解流图

这个分解过程一直进行下去。若取 $N=2^m$,经过 $m-1$ 次分解后,序列已经变成 2 点序列。2 点序列的 DFT 只需要 2 次加法,直接实现,不需要再分解。对于 $N=8$ 的情况,最后的 2 点 DFT 直接用图 3.12 的蝶形图实现,最后得到的完整图形如图 3.16 所示。图 3.16 中最左侧的蝶形中的旋转因子 W_N^0 是为保持图中各级运算单元的一致性加入的。

图 3.16 中,每一级的基本运算单元都是一致的,是一种蝶形运算单元,重新画在图 3.17 中。每个蝶形运算单元需要 1 次乘法和 2 次加法运算。在 $N=2^m$ 的情况下,总共有 m 级运算,每一级由 $N/2$ 个蝶形运算单元构成。故总运算量为

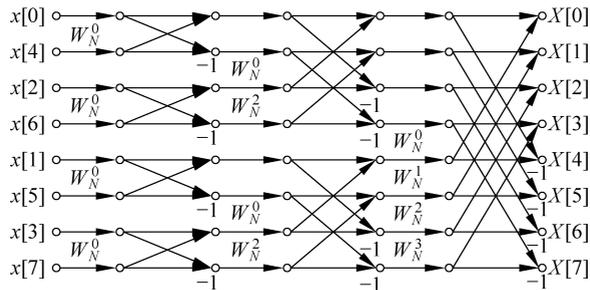


图 3.16 8 点序列按时间抽取 FFT 流图

乘法次数

$$m_c = m \times \frac{N}{2} = \frac{N}{2} \log_2(2^m) = \frac{N}{2} \log_2(N) \quad (3.44)$$

加法次数

$$a_c = m \times N = N \log_2(2^m) = N \log_2(N) \quad (3.45)$$

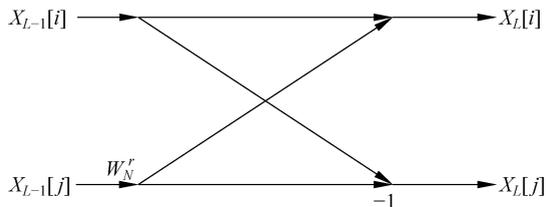


图 3.17 按时间抽取 FFT 的蝶形计算结构

观察图 3.16 的计算流程图,发现图中最左侧信号的输入不再是自然的顺序。图中所示的这种顺序称为倒位序。为了理解倒位序的一般性顺序,需分析序号的二进制表示。设 $N=2^m$ 为有限长信号的长度,需要用 m 位二进制数表示样本的序号 n , n 的二进制表示为

$$n = (b_{m-1}b_{m-2}\cdots b_1b_0)_2 = b_{m-1}2^{m-1} + b_{m-2}2^{m-2} + \cdots + b_12^1 + b_02^0$$

上式中, b_i 只取 0 和 1。序号 n 的倒位序 \bar{n} 记为

$$\bar{n} = (b_0b_1\cdots b_{m-2}b_{m-1})_2 = b_02^{m-1} + b_12^{m-2} + \cdots + b_{m-2}2^1 + b_{m-1}2^0$$

从二进制表示来讲,倒位序是由原序号的二进制表示按位进行次序反转得到。

不难验证,图 3.16 输入端的排列次序是 3 位二进制数的倒位序。例如 $4 = (100)_2$, 其倒位序为 $(001)_2 = 1$,这正是图中 $x[4]$ 的位置。另一例子 $5 = (101)_2$, 其倒位序取值不变。

不难理解为什么按时间抽取 FFT 算法的输入顺序是倒位序排列。在序号的二进制表示中,最低位决定奇偶性,最高位决定前一半和后一半。在按时间抽取的分解时,用最低位(序号的奇偶性)决定了被分到上半还是下半,因此实际上由最低位和最高位进行了交换;在下次分解时,原最高位和最低位都不再起作用,次高位和次低位交换;当分解过程一直进行下去,到最后一级时,二进制位完成了倒位序过程。

注意到,在分解过程中,FFT 的输出顺序保持原来的顺序,称为正位序。

在用数字系统实现 FFT 时,有一些专用处理器内部带有进行倒位序运算的单元,方便 FFT 的编程实现。在用通用计算机编程实现 FFT 运算时,可以通过编写一个专用子程序

巧妙地实现倒位序运算。

3.6.2 按频率抽取基 2-FFT 算法

与时间抽取 FFT 的推导不同,换一种思路,将长序列的前后各一半分开,观察会得到什么结果

$$\begin{aligned} X[k] &= \sum_{n=0}^{\frac{N}{2}-1} x[n]W_N^{nk} + \sum_{n=\frac{N}{2}}^{N-1} x[n]W_N^{nk} \\ &= \sum_{n=0}^{\frac{N}{2}-1} x[n]W_N^{nk} + \sum_{n=0}^{\frac{N}{2}-1} x\left[n + \frac{N}{2}\right]W_N^{\left(n+\frac{N}{2}\right)k} \\ &= \sum_{n=0}^{\frac{N}{2}-1} \left\{x[n] + x\left[n + \frac{N}{2}\right](-1)^k\right\}W_N^{nk} \end{aligned} \quad (3.46)$$

在式(3.46)中,仅考虑 $k=2r$ 的偶数序号 DFT 系数,得到

$$X[2r] = \sum_{n=0}^{\frac{N}{2}-1} \left\{x[n] + x\left[n + \frac{N}{2}\right]\right\}W_N^{rn} = \sum_{n=0}^{\frac{N}{2}-1} e[n]W_{\frac{N}{2}}^{rn} = E[r] \quad (3.47)$$

仅考虑 $k=2r+1$ 的奇数序号 DFT 系数,得到

$$X[2r+1] = \sum_{n=0}^{\frac{N}{2}-1} \left\{x[n] - x\left[n + \frac{N}{2}\right]\right\}W_N^{rn} = \sum_{n=0}^{\frac{N}{2}-1} f[n]W_{\frac{N}{2}}^{rn} = F[r] \quad (3.48)$$

在式(3.47)和式(3.48)中,对偶数序号和奇数序号 DFT 系数的计算,相当于先定义如式(3.49)和式(3.50)所表示的两个 $N/2$ 点长的序列,然后做 $N/2$ 点 DFT,即

$$e[n] = x[n] + x\left[n + \frac{N}{2}\right], \quad 0 \leq n < N/2 \quad (3.49)$$

$$f[n] = \left\{x[n] - x\left[n + \frac{N}{2}\right]\right\}W_N^n, \quad 0 \leq n < N/2 \quad (3.50)$$

以 $N=8$ 为例,这个分解过程如图 3.18 所示。

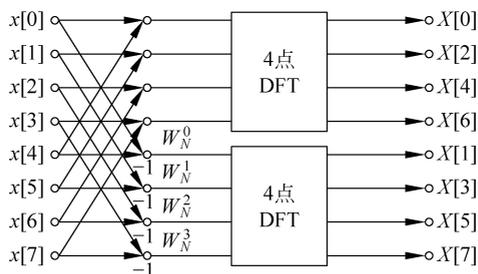


图 3.18 按频率抽取 FFT 的第一次分解流程图

与时间抽取情况类似,第一步分解可节省约一半运算量。可继续分解,第二级分解为

$$a[n] = e[n] + e\left[n + \frac{N}{4}\right], \quad 0 \leq n < N/4$$

$$b[n] = \left\{ e[n] - e \left[n + \frac{N}{4} \right] \right\} W_{N/2}^n = \left\{ e[n] - e \left[n + \frac{N}{4} \right] \right\} W_N^{2n}, \quad 0 \leq n < N/4$$

$$c[n] = f[n] + f \left[n + \frac{N}{4} \right], \quad 0 \leq n < N/4$$

$$d[n] = \left\{ f[n] - f \left[n + \frac{N}{4} \right] \right\} W_{N/2}^n = \left\{ f[n] - f \left[n + \frac{N}{4} \right] \right\} W_N^{2n}, \quad 0 \leq n < N/4$$

显然

$$X[4r] = E[2r] = \sum_{n=0}^{\frac{N}{4}-1} a[n] W_{\frac{N}{4}}^{rn} = A[k]$$

$$X[4r+2] = E[2r+1] = \sum_{n=0}^{\frac{N}{4}-1} b[n] W_{\frac{N}{4}}^{rn} = B[k]$$

$$X[4r+1] = F[2r] = \sum_{n=0}^{\frac{N}{4}-1} c[n] W_{\frac{N}{4}}^{rn} = C[k]$$

$$X[4r+3] = F[2r+1] = \sum_{n=0}^{\frac{N}{4}-1} d[n] W_{\frac{N}{4}}^{rn} = D[k]$$

以 $N=8$ 为例, 第二级分解过程如图 3.19 所示。

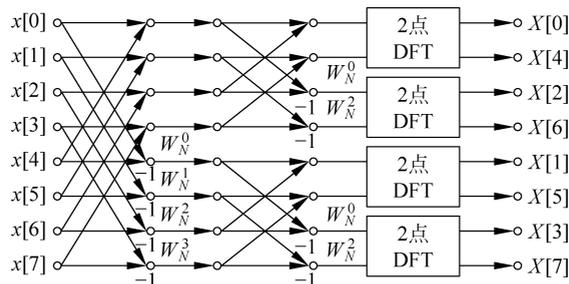


图 3.19 按频率抽取 FFT 的第二次分解流图

这种分解过程一直持续下去, 直到只剩下 2 点长序列, 2 点长序列直接实现即可。以 $N=8$ 为例, 第二级分解后只剩下 2 点序列, 直接用 2 点 DFT 的蝶形实现, 完整的计算过程如图 3.20 所示。

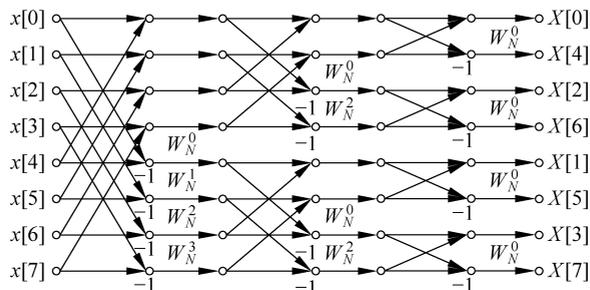


图 3.20 8 点序列按频率抽取 FFT 的流图

由于这种分解算法,在每一步都将 DFT 系数(频域序号)分成偶序号和奇数序号分别运算,因此称为按频率抽取的基 2-FFT 算法。它的基本运算蝶形图如图 3.21 所示,显然其乘法和加法运算量与按时间抽取基 2-FFT 算法一致。

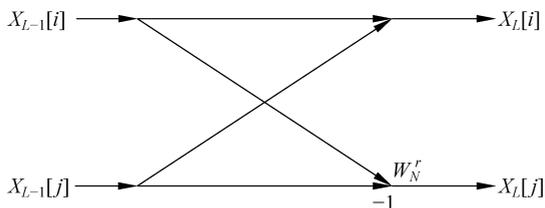


图 3.21 按频率抽取 FFT 的蝶形计算结构

注意到,按频率抽取 FFT 流程中,输入是按正位序排列的,输出是倒位序。

注意到,不管是时间抽取或频率抽取,基 2-FFT 的序号 n 和 k 互为倒序。还要注意到,不管是时间抽取还是频率抽取,蝶形运算结构都是从前一级的两个位置取出数据,经蝶形运算后,结果放置回相同的位置。这种运算结构称为同址运算,可有效节省运算过程中的存储器空间。

3.6.3 基 4 和分裂基 FFT

设 $N=4^m$,类似于基 2 分解的方式讨论基 4-FFT 算法。设 $N=4N_1$,将序列分成 4 个子序列,这里只讨论按时间抽取的基 4-FFT,故按照序号对 4 取余,分成 4 个子序列。

$$n = 4m_0 + n_0 \quad (3.51)$$

这里 $0 \leq m_0 < N_1$, $0 \leq n_0 < 4$, n_0 表示每个子序列, m_0 表示子序列中的序号。

输出序号 k 按下式分解

$$k = N_1 k_{m-1} + \ell_{m-1} \quad (3.52)$$

这个式子是按时间抽取基 2-FFT 输出合成的推广,4 个子序列的 N_1 点 DFT 系数,用序号 ℓ_{m-1} 表示,且 $0 \leq \ell_{m-1} < N_1$,4 个子序列 DFT 系数中,序号 ℓ_{m-1} 取值相等的一组(4 个)值构成一个 4 蝶形,4 蝶形的输出序号用 k_{m-1} , $0 \leq k_{m-1} < 4$ 表示,4 个输出值在最终 DFT 系数列中间隔为 N_1 ,故 DFT 输出序号表示为式(3.52)。

用稍紧凑一点的表示方法,描述第一次分解过程为

$$\begin{aligned} X[k] &= \sum_{n_0=0}^3 \sum_{m_0=0}^{N_1-1} x[4m_0 + n_0] W_N^{(4m_0+n_0)(N_1 k_{m-1} + \ell_{m-1})} \\ &= \sum_{n_0=0}^3 \left\{ \left(\sum_{m_0=0}^{N_1-1} x[4m_0 + n_0] W_{N_1}^{m_0 \ell_{m-1}} \right) W_N^{n_0 \ell_{m-1}} \right\} W_4^{n_0 k_{m-1}} \\ &= \sum_{n_0=0}^3 \{ X[\ell_{m-1}, n_0] W_N^{n_0 \ell_{m-1}} \} W_4^{n_0 k_{m-1}} \end{aligned} \quad (3.53)$$

上式第二行中,小括号内是每个子序列的 DFT,即

$$X[\ell_{m-1}, n_0] = \sum_{m_0=0}^{N_1-1} x[4m_0 + n_0] W_{N_1}^{m_0 \ell_{m-1}}, \quad 0 \leq \ell_{m-1} < N_1$$

针对子序列标号不同,乘以不同的旋转因子 $W_N^{n_0 \ell_{m-1}}$,然后通过 4 蝶形运算合成 N 点 DFT。

图 3.22 是以 $N=64$ 为例的按时间抽取基 4-FFT 的第一次分解的部分流程图。由于 $m=3$,故图中 $\ell_{m-1}=\ell_2$,4 个 16 点 DFT 分别对应 n_0 的 4 个取值,每个 16 点 DFT 输出序号用 ℓ_2 表示。由于复杂性原因,在一张图上已经无法清晰地画出所有蝶形运算,只画出了一个完整 4 蝶形的示意图,图 3.23 单独画出了一个标准 4 蝶形运算。

与基 2 一样,式(3.53)中的每一个 N_1 点 DFT 可继续分解,直到分解为 4 点为止。通过对每一级分解和每一个 4 蝶形运算量的统计,可以得到基 4-FFT 的乘法运算量为

$$\begin{aligned} m_c &= 3 \times \frac{N}{4} \times (m-1) \approx \frac{3}{4} Nm = \frac{3}{4} N \log_4 N \\ &= \frac{3}{4} N \left(\frac{1}{2} \log_2 N \right) = \frac{3}{4} \left(\frac{N}{2} \log_2 N \right) \end{aligned}$$

可见,基 4-FFT 比基 2-FFT 进一步减少乘法运算次数。

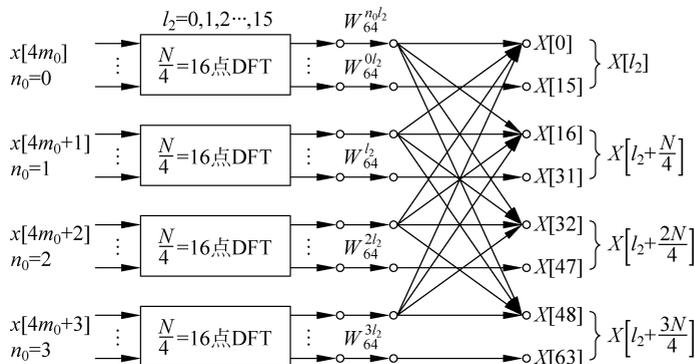


图 3.22 $N=64$ 的基 4-FFT 部分流程图

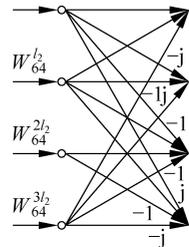


图 3.23 4 蝶形运算流程图

直接实现 4 蝶形需要 12 次加法,故加法运算量为

$$a_c = 12 \times \frac{N}{4} \times (m-1) \approx 3Nm = 3N \log_4 N = \frac{3}{2} N \log_2 N$$

可见,基 4-FFT 的加法运算量高于基 2-FFT。若对每一个基 4 蝶形重新安排一下运算(分解成 2 蝶形的级联,如同 4 点序列的基 2-FFT 实现,见习题)则只需 8 次加法,这样基 4-FFT 的加法运算与基 2-FFT 的相等。

分裂基 分裂基是基 2 和基 4 两种 FFT 的结合。尽管 1965 年就提出了基 2-FFT,基 4-FFT 是基 2 的直接推广,但分裂基却晚了近 20 年。直到 1984 年由杜阿梅尔(P Dohamel)和霍尔曼(H Hollmann)提出。

在按时间抽取基 2-FFT 的算法流程中,偶数序号构成的子序列产生的 DFT 系数没有加旋转因子,奇数序号子序列的 DFT 系数都加了旋转因子。由于基 4-FFT 进一步减少乘法运算,以此为启发,将奇数序号进一步再分成两个子序列,这样就构成了 3 个子序列,即

$$x_1[r] = x[2r], \quad 0 \leq r \leq \frac{N}{2} - 1$$

$$x_2[l] = x[4l+1], \quad 0 \leq l \leq \frac{N}{4} - 1$$

$$x_3[l] = x[4l + 3], \quad 0 \leq l \leq \frac{N}{4} - 1$$

将原序列的 DFT 分解成三个子序列的 DFT 的组合, 进行类似基 2 或基 4 的推导得到

$$X[k] = X_1[(k)_{N/2}] + W_N^k X_2[(k)_{N/4}] + W_N^{3k} X_3[(k)_{N/4}], \quad 0 \leq k \leq N - 1$$

将上式分成 4 段, 针对 $0 \leq k \leq \frac{N}{4} - 1$ 得到

$$\begin{aligned} X[k] &= X_1[k] + W_N^k X_2[k] + W_N^{3k} X_3[k] \\ X\left[k + \frac{N}{4}\right] &= X_1\left[k + \frac{N}{4}\right] + W_N^{(k+\frac{N}{4})} X_2\left[\left(\left(k + \frac{N}{4}\right)\right)_{\frac{N}{4}}\right] + W_N^{3(k+\frac{N}{4})} X_3\left[\left(\left(k + \frac{N}{4}\right)\right)_{\frac{N}{4}}\right] \\ &= X_1\left[k + \frac{N}{4}\right] - j(W_N^k X_2[k] - W_N^{3k} X_3[k]) \\ X\left[k + \frac{N}{2}\right] &= X_1\left[\left(\left(k + \frac{N}{2}\right)\right)_{\frac{N}{2}}\right] + W_N^{(k+\frac{N}{2})} X_2\left[\left(\left(k + \frac{N}{2}\right)\right)_{\frac{N}{4}}\right] \\ &\quad + W_N^{3(k+\frac{N}{2})} X_3\left[\left(\left(k + \frac{N}{2}\right)\right)_{\frac{N}{4}}\right] \\ &= X_1[k] - W_N^k X_2[k] - W_N^{3k} X_3[k] \\ X\left[k + \frac{3N}{4}\right] &= X_1\left[\left(\left(k + \frac{3N}{4}\right)\right)_{\frac{N}{2}}\right] + W_N^{(k+\frac{3N}{4})} X_2\left[\left(\left(k + \frac{3N}{4}\right)\right)_{\frac{N}{4}}\right] \\ &\quad + W_N^{3(k+\frac{3N}{4})} X_3\left[\left(\left(k + \frac{3N}{4}\right)\right)_{\frac{N}{4}}\right] \\ &= X_1\left[k + \frac{N}{4}\right] + j(W_N^k X_2[k] - W_N^{3k} X_3[k]) \end{aligned}$$

稍加整理, 重新写为

$$\begin{cases} X[k] = X_1[k] + W_N^k X_2[k] + W_N^{3k} X_3[k], \\ X\left[k + \frac{N}{2}\right] = X_1[k] - (W_N^k X_2[k] + W_N^{3k} X_3[k]), \\ X\left[k + \frac{N}{4}\right] = X_1\left[k + \frac{N}{4}\right] - j(W_N^k X_2[k] - W_N^{3k} X_3[k]), \\ X\left[k + \frac{3N}{4}\right] = X_1\left[k + \frac{N}{4}\right] + j(W_N^k X_2[k] - W_N^{3k} X_3[k]), \end{cases} \quad 0 \leq k \leq \frac{N}{4} - 1 \quad (3.54)$$

图 3.24 给出分裂基实现的流图。与基 4 情况一样, 图中只完整画出了一个蝶形, 分裂基的蝶形结构更一般地表示在图 3.25 中。

注意到分裂基的特点, 它的蝶形是倒 L 型的, 同时存在不同点数的 DFT 需要进一步分解。分裂基运算量的统计留作习题, 乘法和加法运算量的近似结果分别为

$$m_c = \frac{1}{3} N \log_2 N$$

$$a_c = N \log_2 N$$

分裂基的乘法运算量比基 2-FFT 节省 33%, 加法运算量相同。在所讨论的基 2、基 4 和分裂基 FFT 中, 分裂基运算量最少, 实现结构比基 2-FFT 稍复杂。分裂基要求 $N = 2^m$ 而不是基 4 的 $N = 4^m$, 对序列长度的要求与基 2 相同, 比基 4 算法灵活。综合来讲, 分裂基在算

法有效性和复杂性方面取得了好的平衡。

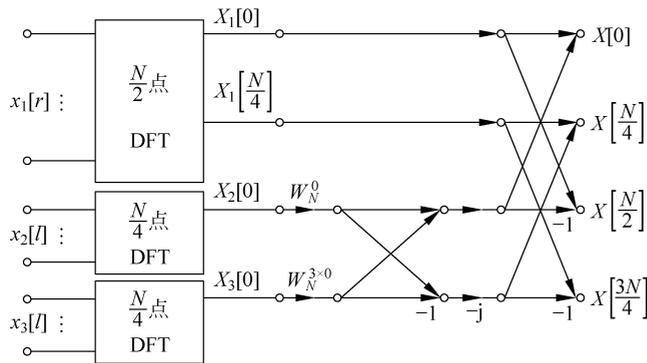


图 3.24 分裂基实现流程图

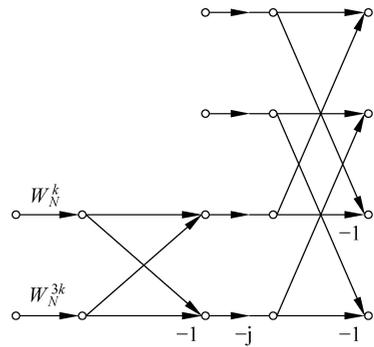


图 3.25 分裂基的蝶形结构

3.6.4 滑窗 FFT 算法

如果数据做 DFT 时是滑动处理的,即每次处理之后,观察窗移动一个采样点。起始做如下序列的 FFT

$$x[0], x[1], x[2], \dots, x[N-1]$$

滑动一步,做如下序列的 FFT

$$x[1], x[2], \dots, x[N-1], x[N]$$

⋮

滑动 m 步,做如下序列的 FFT

$$x[m], x[m+1], \dots, x[m+N-2], x[m+N-1]$$

现假设已计算出第 m 步的 DFT 结果 $X_m[k]$,求下次滑动后的 DFT 结果。根据 DFT 的定义,第 m 步的 DFT 为

$$X_m[k] = \sum_{n=0}^{N-1} x[m+n] W_N^{nk}$$

第 $m+1$ 步的 DFT 为

$$\begin{aligned} X_{m+1}[k] &= \sum_{n=0}^{N-1} x[m+1+n] W_N^{nk} \\ &= \sum_{n=-1}^{N-2} x[m+1+n] W_N^{nk} + x[m+N] W_N^{-k} - x[m] W_N^{-k} \\ &= W_N^{-k} \sum_{n'=0}^{N-1} x[m+n'] W_N^{n'k} + x[m+N] W_N^{-k} - x[m] W_N^{-k} \\ &= W_N^{-k} \{ X_m[k] + x[m+N] - x[m] \} \end{aligned} \quad (3.55)$$

可见滑动一步之后的 DFT 可从上一步的 DFT 结果加上滑进窗内的新数据与滑出窗外的老数据之差,再乘以旋转因子 W_N^{-k} 得到。

滑动 DFT 算法的优点是计算量比直接做 FFT 还要小。如果起始完成一次 FFT(假设采用基 2 算法)之后滑动 M 步,则总的复数乘法计算量为

$$\frac{1}{2}N\log_2 N + MN$$

平均每一步滑动 DFT 的计算量为

$$\frac{N}{2M}\log_2 N + N$$

当 $M \approx N$ 时, 平均每步 DFT 的计算量为 $O(N)$ 。

滑动 DFT 的另一个优点是, 滑动过程中可以不必计算所有频点的变换结果, 而只计算所关心的频点的 DFT 结果, 这对某些应用而言可以进一步减少计算量。

以上讨论了每次只滑动一步, 若滑动 M 步则由 M 次一步滑动构成, 也可以讨论一次直接滑动 M 步的情况, 其详细推导留作练习。

由于滑动过程实际上是一个积累过程, 因此滑动 DFT 存在误差稳定性的问题。一旦由于某种原因在滑动计算过程中引入某个误差, 这个误差将保持在滑动计算结果中, 直至下次重新计算一次起始 FFT 才能将这个误差清除。

* 3.6.5 组合数 FFT 算法简述

将 FFT 的分解思想应用到任意组合数的情况。假定 N 是一个组合数, 可以表达成 v 个因子的乘积

$$N = N_1 N_2 \cdots N_v$$

例如, $N = 3 \times 2 \times 5$ 是一个组合数的例子。

为了简化表示, 用 $N_{a \sim b}$ 表示下标从 a 到 b 的连乘, 即

$$N_{a \sim b} = N_a \cdots N_b$$

这样

$$N = N_{1 \sim v}$$

则任意 $n = 0, 1, \dots, N-1$ 可以表示为

$$n = n_0 + n_1 N_1 + n_2 N_{1 \sim 2} + \cdots + n_{v-1} N_{1 \sim v-1}$$

其中

$$\begin{aligned} n_0 &= 0, 1, \dots, N_1 - 1 \\ n_1 &= 0, 1, \dots, N_2 - 1 \\ &\vdots \\ n_{v-1} &= 0, 1, \dots, N_v - 1 \end{aligned}$$

任意 $k = 0, 1, \dots, N-1$ 可以表示为

$$k = k_0 + k_1 N_v + k_2 N_{v \sim v-1} + \cdots + k_{v-1} N_{v \sim 2}$$

其中

$$\begin{aligned} k_0 &= 0, 1, \dots, N_v - 1 \\ k_1 &= 0, 1, \dots, N_{v-1} - 1 \\ &\vdots \\ k_{v-1} &= 0, 1, \dots, N_1 - 1 \end{aligned}$$

DFT 的变换基可以表示为

$$W_N^{nk} = W_N^{(n_0 + n_1 N_1 + n_2 N_{1 \sim 2} + \cdots + n_{v-1} N_{1 \sim v-1})k}$$

$$\begin{aligned}
&= W_N^{n_0 k} W_N^{n_1 N_1 k} W_N^{n_2 N_{1 \sim 2} k} \cdots W_N^{n_{v-1} N_{1 \sim v-1} k} \\
&= W_N^{n_0 k} W_{N/N_1}^{n_1 k} W_{N/N_{1 \sim 2}}^{n_2 k} \cdots W_{N/N_{1 \sim v-1}}^{n_{v-1} k} \\
&= W_{N_{1 \sim v}}^{n_0 k} W_{N_{2 \sim v}}^{n_1 k} W_{N_{3 \sim v}}^{n_2 k} \cdots W_{N_v}^{n_{v-1} k}
\end{aligned}$$

根据 DFT 的定义

$$\begin{aligned}
X[k] &= \sum_{n=0}^{N-1} x[n = n_0 + n_1 N_1 + \cdots + n_{v-1} N_{1 \sim v-1}] W_N^{nk} \\
&= \sum_{n_0=0}^{N_1-1} W_{N_{1 \sim v}}^{n_0 k} \sum_{n_1=0}^{N_2-1} W_{N_{2 \sim v}}^{n_1 k} \cdots \sum_{n_{v-1}=0}^{N_v-1} x[n] W_{N_v}^{n_{v-1} k}
\end{aligned}$$

可见, DFT 可以分 v 步逐步计算。第一步令

$$G_{k_0}[n'] = \sum_{n_{v-1}=0}^{N_v-1} x[n' + n_{v-1} N_{1 \sim v-1}] W_{N_v}^{n_{v-1} k_0}, \quad n' = 0, 1, \dots, N_{1 \sim v-1} - 1$$

这样的子序列一共有 N_v 个, 每个序列的长度为 $N_{1 \sim v-1}$ 。对每个这样的子序列再进行下一步计算, 令

$$G_{k_1 k_0}[n'] = \sum_{n_{v-2}=0}^{N_{v-1}-1} G_{k_0}[n' + n_{v-2} N_{1 \sim v-2}] W_{N_{v \sim v-1}}^{n_{v-2}(k_0 + k_1 N_v)}, \quad n' = 0, 1, \dots, N_{1 \sim v-2} - 1$$

对每一个 k_0 , 这样的子序列共有 $N_{v \sim v-1}$ 个, 每个序列的长度为 $N_{1 \sim v-2}$ 。如此重复计算, 第 m 步有

$$G_{k_{m-1} \cdots k_1 k_0}[n'] = \sum_{n_{v-m}=0}^{N_{v-m+1}-1} G_{k_{m-2} \cdots k_1 k_0}[n' + n_{v-m} N_{1 \sim v-m}] W_{N_{v \sim v-m+1}}^{n_{v-m} k}, \quad n' = 0, 1, \dots, N_{1 \sim v-m} - 1$$

对每个给定的 $k_{m-2} \cdots k_1 k_0$, 这样的子序列共有 $N_{v \sim v-m+1}$ 个, 每个序列的长度为 $N_{1 \sim v-m}$ 。如此重复计算直至子序列长度为 1 止, 一共要计算 v 步。每一步蝶形的复数乘法次数为 $N_{v \sim v-m+1} N_{1 \sim v-m} N_{v-m+1} = N N_{v-m+1}$, 复数加法次数为

$$N_{1 \sim v-m} N_{v \sim v-m+1} (N_{v-m+1} - 1) = N (N_{v-m+1} - 1)$$

故总的计算量为: 复数乘法

$$\sum_{m=1}^v (N N_{v-m+1}) = N \sum_{m=0}^{v-1} N_{v-r} = N \sum_{m=1}^v N_m$$

复数加法

$$\sum_{m=1}^v N (N_m - 1)$$

计算量比直接计算 DFT 所需的 N^2 要少很多, 且 k 和 n 是位 (digit) 倒序关系 (称为广义倒序关系)。

3.6.6 快速傅里叶反变换

对于 IDFT, 导出快速算法的思路与 DFT 几乎一致。由于反变换和正变换的计算结构如此相似, 不再单独导出快速傅里叶反变换 (IFFT) 算法, 而是采用 FFT 实现 IFFT。由 IDFT 的定义

$$x[n] = \text{IDFT}\{X[k]\} = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn} = \frac{1}{N} \left(\sum_{k=0}^{N-1} X^* [k] W_N^{kn} \right)^*$$

上式中,括号内的运算与 DFT 运算完全一致,只是需要以 $X^*[k]$ 为输入序列,故以 FFT 运算单元实现 IFFT 的过程如下:

- (1) 以 $X^*[k]$ 作为输入,执行 FFT 运算;
- (2) FFT 输出取共轭并除以 N ,得到反变换 $x[n]$ 。

3.7 CZT 算法

DFT 和其标准 FFT 算法都有其局限性。当 DFT 的输入 N 点,输出 N 点,DFT 系数均匀分布于 z 平面单位圆,两个系数的频域间隔为 $\frac{2\pi}{N}$ 。可见,DFT 有如下局限性:

(1) 当输入采样点数很少时,若希望 DFT 输出的频率点数较多,则需要补零做大点数 FFT,增加了计算量;

(2) 对于窄带信号,希望通带内采样点密,带外疏,或根本不用计算,用 DFT 计算需计算全部密集采样点,再取所需的一部分;

(3) 难以准确得到信号的“自然频率”位置,DFT 系数代表的是信号频谱在 $\frac{2\pi}{N}k$,而不是任意感兴趣点的取值。

例 3.7.1 设一个带通信号,采样获得了 $N=150$ 个点,希望只分析在 $\frac{\pi}{4} \sim \frac{3\pi}{8}$ 之间的频谱,希望在这 $\frac{\pi}{8}$ 的间隔内,获得 128 点的采样密度,即两个 DFT 系数对应的频率间隔为 $\frac{\pi}{8 \times 128}$,如果用 DFT(通过 FFT 计算),需要计算的总点数是 $L=128 \times \frac{2\pi}{\pi/8}=2048$,由于采样点只有 150 点,因此,需要补上 $L-N=1898$ 个零,然后做 $L=2048$ 点 FFT,FFT 结束后取出相当于频率 $\frac{\pi}{4} \sim \frac{3\pi}{8}$ 的 128 个系数,即 $X[256] \sim X[384]$ 的点。

CZT 变换相当于是一种广义的 DFT,同时也可以进一步利用快速卷积方法给出 CZT 的快速算法。相比于 DFT 是对 z 变换在单位圆上的均匀采样,CZT 变换相当于是在 z 平面一个更一般曲线上的采样值。如图 3.26,左侧是 DFT 在 z 平面的部分采样点,右图是 CZT 在 z 平面的部分采样点,CZT 的采样点是位于一条螺旋线上的。

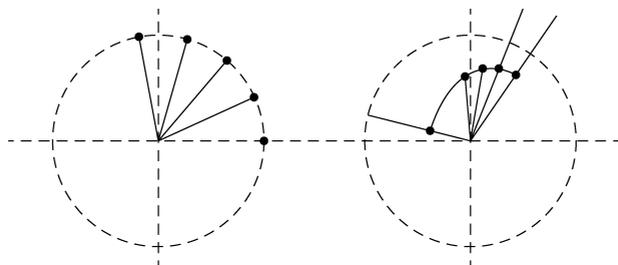


图 3.26 DFT(左)和 CZT(右)取样点比较

对于 N 点序列, CZT 计算 M 个输出, N 和 M 可以相等, 也可以不等。CZT 的一般定义为

$$X_{CZ}(z) \Big|_{z=z_k} = \sum_{n=0}^{N-1} x[n] z_k^{-n}, \quad k=0, 1, \dots, M-1 \quad (3.56)$$

其中

$$z_k = AW^{-k}, \quad 0 \leq k \leq M-1$$

更细致地取

$$z_k = (A_0 e^{j\theta_0}) (W_0 e^{-j\phi_0})^{-k}$$

实际上 z_k 中的 $A=A_0 e^{j\theta_0}$ 代表第一个采样点在 z 平面的位置, $W=W_0 e^{-j\phi_0}$ 表示每一个采样点在 z 平面的步进关系, ϕ_0 表示两个采样点对应的角度差, W_0 表示采样曲线的形式。 $W_0 > 1$, 螺线向中心弯曲; $W_0 < 1$, 螺线远离中心弯曲; 若取 $W_0 = 1, A_0 = 1$, 则采样点位于单位圆上, 但起始点由 θ_0 确定。

例 3.7.2 若取 $\theta_0 = 0, \phi_0 = \frac{2\pi}{N}, W_0 = A_0 = 1, N = M$, 显然 CZT 就退化为 DFT; 若取 $\theta_0 = \frac{\pi}{4}, \phi_0 = \frac{2\pi}{2048}, W_0 = A_0 = 1, N = 150, M = 128$, 用 CZT 即可只计算例 3.7.1 要求的 $\frac{\pi}{4} \sim \frac{3\pi}{8}$ 之间的频谱。

如下导出利用卷积计算 CZT 的算法, 由

$$X_{CZ}(z_k) = \sum_{n=0}^{N-1} x[n] z_k^{-n} = \sum_{n=0}^{N-1} x[n] A^{-n} W^{nk}, \quad 0 \leq k \leq M-1$$

利用

$$nk = \frac{1}{2} [n^2 + k^2 - (k-n)^2]$$

代入上式得到

$$\begin{aligned} X_{CZ}(z_k) &= \sum_{n=0}^{N-1} x[n] A^{-n} W^{\frac{1}{2}n^2} W^{-\frac{1}{2}(k-n)^2} W^{\frac{1}{2}k^2} \\ &= W^{\frac{1}{2}k^2} \sum_{n=0}^{N-1} (x[n] A^{-n} W^{\frac{1}{2}n^2}) W^{-\frac{1}{2}(k-n)^2} \end{aligned} \quad (3.57)$$

如果定义两个新序列

$$f[n] = x[n] A^{-n} W^{\frac{1}{2}n^2}, \quad 0 \leq n < N \quad (3.58)$$

$$h[n] = W^{-\frac{1}{2}n^2}, \quad -(N-1) \leq n \leq M-1 \quad (3.59)$$

代入 CZT 表示式得

$$X_{CZ}(z_k) = W^{\frac{1}{2}k^2} \sum_{n=0}^{N-1} f[n] h[k-n] = W^{\frac{1}{2}k^2} f[k] * h[k] \quad (3.60)$$

此式将 CZT 表示成两个序列的卷积形式。以卷积方式实现 CZT 计算的方框图如图 3.27 所示。

由于两个有限长序列的卷积可用 DFT 来计算, 故可用 FFT 实现 CZT 的卷积计算形

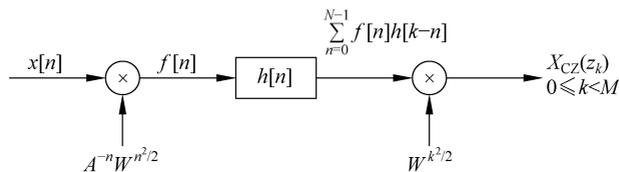


图 3.27 CZT 计算的卷积结构

式,若取 FFT 的长度 $L \geq N + M - 1$,通过周期延拓得到 $h[n]$ 在 $0 \leq n \leq L - 1$ 的取值,如图 3.28 所示。

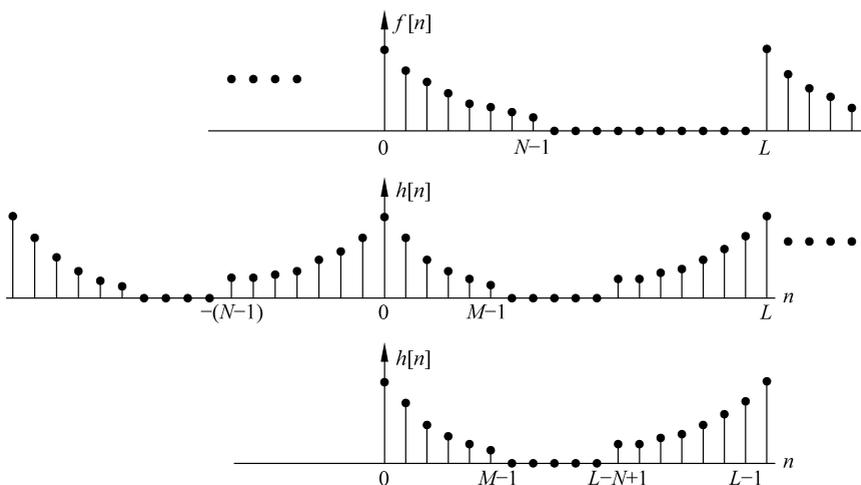


图 3.28 单位抽样响应的取值范围

用 FFT 计算 CZT 的方框图如图 3.29 所示,算法详细表述为如下步骤:

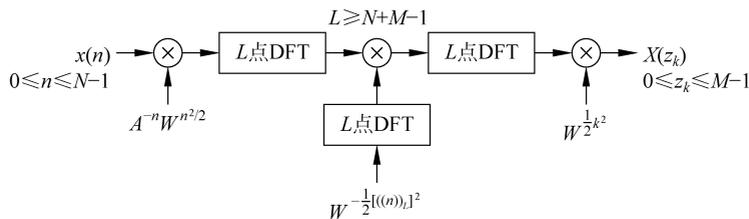


图 3.29 CZT 的 FFT 实现结构

(1) 选择可用 FFT 计算的点数,取 $L = 2^m$;

(2) 计算 $f[n] = \begin{cases} x[n] [A^{-n} W^{\frac{n^2}{2}}], & 0 \leq n < N \\ 0, & N \leq n < L \end{cases}$; (N 次乘法)

(3) 计算 $\text{DFT}\{f[n]\} = F[k]$; ($\frac{1}{2}L \log_2 L$ 次乘法)

(4) 计算

$$h[n] = \begin{cases} W^{-n^2/2}, & 0 \leq n \leq M-1 \\ 0, & M \leq n < L-N \\ W^{-(L-n)^2/2}, & L-N+1 \leq n < L \end{cases};$$

(5) 计算 $\text{DFT}\{h[n]\} = H[k], 0 \leq k \leq L-1$;

(6) 计算 $H[k]F[k]$; (L 次乘法)

(7) 计算 $y[k] = \text{IDFT}\{H[k]F[k]\}$; ($\frac{1}{2}L \log_2 L$ 次乘法)

(8) 取 $y[k]$ 的 L 点中的前 M 点并乘权值: $X_{\text{CZ}}(z_k) = W^{\frac{1}{2}k^2} y[k], 0 \leq k \leq M-1$ 。
(M 次乘法)

在该算法实现中,由于 $h[n]$ 是预先已知的量,因此(4)和(5)两步可预先计算好,在实际操作时不需要运算量。其他各步的运算量放在算法表述的括号中。注意到,这里只统计了乘法次数,并假设用的是基 2-FFT,若采用分裂基等其他 FFT 算法可相应调整系数。集中起来总乘法运算量为

$$\begin{aligned} m_{\text{cf}} &= N + \frac{1}{2}L \log_2 L + L + \frac{1}{2}L \log_2 L + M \\ &= L(\log_2 L + 1) + M + N \end{aligned} \quad (3.61)$$

不难统计,直接用卷积计算 CZT,需要乘法运算量

$$m_{\text{cc}} = NM \quad (3.62)$$

例 3.7.3 对例 3.7.1 的计算任务,用 CZT 完成,如例 3.7.2 讨论的, $N=150, M=128$,用直接卷积实现 CZT,需要运算量

$$m_{\text{cc}} = 150 \times 128 = 19\,200$$

若使用 FFT 计算 CZT,利用基 2-FFT 算法,故取 $L=512$,得到运算量

$$m_{\text{cc}} = 512 \times (9+1) + 150 + 128 = 5398$$

若采用例 3.7.1 所示的 2048 点 FFT,乘法运算量为

$$m_{\text{c}} = 2048 \times 11/2 = 11\,264$$

在这个任务中,利用 FFT 的 CZT 算法,运算量最低。

3.8 离散余弦变换及其快速算法

除了 DFT 外,数字信号处理领域还存在多种离散正交变换,其中有些变换也得到了重要应用。例如 DCT 变换特别适用于信号压缩问题,在当前的数字图像和数字视频压缩标准中被广泛采用。

3.8.1 离散余弦变换

离散余弦变换(DCT)广泛应用于信息压缩中,一维 DCT 变换的基函数集是

$$a_k[n] = \begin{cases} \frac{1}{\sqrt{N}}, & k=0, 0 \leq n \leq N-1 \\ \sqrt{\frac{2}{N}} \cos \frac{\pi(2n+1)k}{2N}, & 1 \leq k \leq N-1, 0 \leq n \leq N-1 \end{cases} \quad (3.63)$$

因此, 矢量 $\mathbf{x} = [x[0], x[1], \dots, x[N-1]]^T$ 的一维 DCT 定义为

$$X_{\text{DCT}}[k] = d[k] \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x[n] \cos \left[\frac{\pi(2n+1)k}{2N} \right], \quad 0 \leq k \leq N-1 \quad (3.64)$$

这里

$$\begin{cases} d[0] = 1 \\ d[k] = \sqrt{2}, \quad 1 \leq k \leq N-1 \end{cases}$$

反变换(IDCT)为

$$x[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} d[k] X_{\text{DCT}}[k] \cos \left[\frac{\pi(2n+1)k}{2N} \right] \quad (3.65)$$

很容易验证式(3.63)的基函数集满足正交性和完备性, 因此式(3.64)和式(3.65)构成正交变换对。DCT 的由来还可以从几个方面来理解。DCT 可看作 KL 变换的一种逼近, KL 变换可去除变换序列中的相关性, 是信源编码中最理想的变换。但 KL 变换实现复杂, 基函数集不定, 限制了其实际应用。人们证明, DCT 变换是 KL 变换的一种良好的逼近。正是如此, DCT 在图像和视频编码标准中被广泛采用。由于 DFT 变换的物理概念明确, 接下来讨论 DCT 与 DFT 的关系, 以此帮助理解 DCT 的物理意义。

1. DCT 与 DFT 的关系

可以通过序列延拓建立 DFT 和 DCT 之间的关系, 对于 N 点序列 $x[n]$, 通过对称性延拓得到序列 $x_2[n]$ 为

$$x_2[n] = \begin{cases} x[n], & 0 \leq n < N \\ x[2N-n-1], & N \leq n < 2N-1 \end{cases} \quad (3.66)$$

序列 $x_2[n]$ 是长度 $2N$ 的对称性序列。图 3.30 示出了 $N=5$ 的序列对称延拓成 $N=10$ 序列的例子。

对 $x_2[n]$ 做 $2N$ 点 DFT 得到

$$\begin{aligned} X_2[k] &= \sum_{n=0}^{2N-1} x_2[n] W_{2N}^{kn} \\ &= \sum_{n=0}^{N-1} x[n] W_{2N}^{kn} + \sum_{n=N}^{2N-1} x[2N-n-1] W_{2N}^{kn} \end{aligned}$$

对第二个求和项做变量替换为 $n' = 2N - n - 1$, 得

$$X_2[k] = \sum_{n=0}^{N-1} x[n] W_{2N}^{kn} + \sum_{n'=N-1}^0 x[n'] W_{2N}^{k(2N-n'-1)}$$

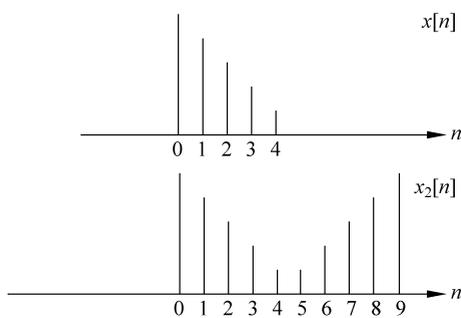


图 3.30 延拓的对称序列

$$\begin{aligned}
&= \sum_{n=0}^{N-1} x[n] [e^{-j\frac{\pi}{2N} \cdot 2kn} + e^{j\frac{\pi}{2N} k(2n+2)}] \\
&= \sum_{n=0}^{N-1} x[n] e^{j\frac{\pi}{2N} k} [e^{-j\frac{\pi}{2N} k(2n+1)} + e^{j\frac{\pi}{2N} k(2n+1)}] \\
&= 2e^{j\frac{\pi}{2N} k} \sum_{n=0}^{N-1} x[n] \cos \frac{\pi k(2n+1)}{2N}
\end{aligned}$$

对比式(3.64)有

$$X_2[k] = \begin{cases} 2\sqrt{N}X_{\text{DCT}}[0], & k=0 \\ \sqrt{2N}e^{j\frac{\pi}{2N}k}X_{\text{DCT}}[k], & 0 < k < N \end{cases} \quad (3.67)$$

或由 $2N$ 点 DFT 系数 $X_2[k]$ 表示 DCT 系数为

$$\begin{aligned}
X_{\text{DCT}}[k] &= \begin{cases} \frac{1}{2\sqrt{N}}X_2[0], & k=0 \\ \frac{1}{\sqrt{2N}}e^{-j\frac{\pi}{2N}k}X_2[k], & 0 < k < N \end{cases} \\
&= \delta[k] \frac{1}{2\sqrt{N}}e^{-j\frac{\pi}{2N}k}X_2[k], \quad 0 \leq k < N \quad (3.68)
\end{aligned}$$

除了比例因子外, N 点序列的 DCT 系数可由对称延拓 $2N$ 点序列 DFT 的前 N 个系数得到。也可以把 DCT 看成是 $2N$ 点变换, 利用式(3.64)可验证(留作习题)

$$X_{\text{DCT}}[k] = -X_{\text{DCT}}[2N - k], \quad 0 \leq k < 2N$$

并且把式(3.67)推广为

$$X_2[k] = \begin{cases} 2\sqrt{N}X_{\text{DCT}}[0], & k=0 \\ \sqrt{2N}e^{j\frac{\pi}{2N}k}X_{\text{DCT}}[k], & 0 < k < N \\ 0, & k=N \\ -\sqrt{2N}e^{j\frac{\pi}{2N}k}X_{\text{DCT}}[2N - k], & N \leq k < 2N \end{cases} \quad (3.69)$$

因为 DFT 意味着时域的周期延拓, 故 $2N$ 点对称序列的 DFT 其实质对应了如图 3.31 所示的周期延拓序列。这种延拓相当于先做对称延拓, 再以 $2N$ 为周期进行周期延拓, 与直接对序列 $x[n]$ 以 N 为周期进行周期延拓相比, 大大减少了周期延拓后序列在周期边界处的跳跃性。因此, $X_2[k]$ 随 k 的衰减要快于直接对 $x[n]$ 的 DFT 变换 $X[k]$, 由此使得 $X_{\text{DCT}}[k]$ 比 $X[k]$ 随 k 的衰减更快。关于这方面的性质, 在 DCT 的能量紧致特性中进一步讨论。

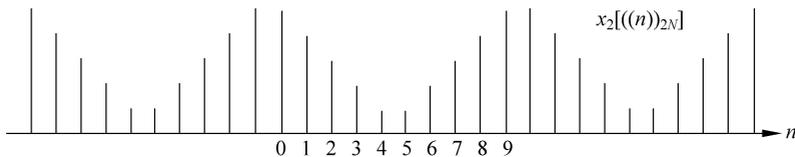


图 3.31 对称延拓序列的周期延拓

2. DCT 的能量紧致性

对于有限长序列的正交变换,变换系数和信号序列是完全等价的,由一种表示可精确得到另外一种表示。但在一些应用中,需要用到变换的一种特殊性质,即能量紧致性。所谓能量紧致性指的是用变换系数的子集有效重构信号序列的能力。

信号紧致性的典型应用是信源的压缩编码,例如语音、图像或视频的压缩编码,用尽可能少的数据尽可能精确地恢复信号是信源编码的目标。目前最常用的编码技术是变换编码。首先对信号进行变换,在变换域只保留部分变换系数,对这部分变换系数进行存储或传输,在播放端或接收端用这些部分系数反变换重构信号。由于只保留部分变换系数,重构的信号必定存在误差。出于对信源质量的要求,对误差大小要有限制。也就是给定一个误差门限,在误差值不大于门限的情况下,保留最少的变换系数。

对于一种变换来讲,给出要求的误差门限,需要保留的系数数目越少,则称该变换的能量紧致性越好。人们发现,对于大多数实际信号,DCT 变换的能量紧致性比 DFT 要好。也正是这个原因,图像和视频编码标准大都采用 DCT 变换。

设有 N 点有限长序列 $x[n]$,分别进行 N 点 DFT 和 DCT,系数分别记为 $X[k]$ 和 $X_{\text{DCT}}[k]$,用 Π 表示保留的 DFT 系数的序号集合,用 Π_{DCT} 表示保留的 DCT 系数的序号集合,用 $|\Pi|$ 表示集合中元素的个数,用 $\hat{X}[k]$ 和 $\hat{X}_{\text{DCT}}[k]$ 表示只保留部分系数,其他系数置为零,即

$$\hat{X}[k] = \begin{cases} X[k], & k \in \Pi \\ 0, & k \notin \Pi \end{cases} \quad (3.70)$$

和

$$\hat{X}_{\text{DCT}}[k] = \begin{cases} X_{\text{DCT}}[k], & k \in \Pi_{\text{DCT}} \\ 0, & k \notin \Pi_{\text{DCT}} \end{cases} \quad (3.71)$$

利用部分 DFT 系数重构的信号为

$$\hat{x}[n] = \text{IDFT}\{\hat{X}[k]\} \quad (3.72)$$

利用部分 DCT 系数重构的信号为

$$\hat{x}_{\text{DCT}}[n] = \text{IDCT}\{\hat{X}[k]\} \quad (3.73)$$

重构信号的平方误差和分别为

$$e = \sum_{n=0}^{N-1} (x[n] - \hat{x}[n])^2 \quad (3.74)$$

和

$$e_{\text{DCT}} = \sum_{n=0}^{N-1} (x[n] - \hat{x}_{\text{DCT}}[n])^2 \quad (3.75)$$

对于一般实际信号(实际语音、实际图像或视频数据)DCT 比 DFT 的能量紧致性好是指如下两种情况之一:若 $|\Pi_{\text{DCT}}| = |\Pi|$,则 $e_{\text{DCT}} < e$; 或取 $e_{\text{DCT}} = e$,则有 $|\Pi_{\text{DCT}}| < |\Pi|$ 。

与帕塞瓦尔定理类似,可以证明误差的时域和频域等价表示,即如下的两个等式,证明过程留作习题。

$$e[\Pi] = \sum_{n=0}^{N-1} (x[n] - \hat{x}[n])^2 = \frac{1}{N} \left(\sum_{k=0}^{N-1} |X[k]|^2 - \sum_{k \in \Pi} |X[k]|^2 \right)$$

$$= \frac{1}{N} \sum_{k \notin \Pi} |X[k]|^2 \quad (3.76)$$

和

$$\begin{aligned} e_{\text{DCT}}[\Pi_{\text{DCT}}] &= \sum_{n=0}^{N-1} (x[n] - \hat{x}_{\text{DCT}}[n])^2 = \sum_{k=0}^{N-1} |X_{\text{DCT}}[k]|^2 - \sum_{k \in \Pi_{\text{DCT}}} |X_{\text{DCT}}[k]|^2 \\ &= \sum_{k \in \Pi_{\text{DCT}}} |X_{\text{DCT}}[k]|^2 \end{aligned} \quad (3.77)$$

注意以上两式的区别, DCT 的基序列是归一化的, DFT 不是归一化的, 故需要除以 N 。

在变换编码中, 怎样选择保留系数的集合 Π 和 Π_{DCT} 是有技巧的, 进一步的细节讨论超出本书范围, 有兴趣的读者可参考有关逼近论或信源编码的文献[30, 58]。

例 3.8.1 对于一个 $N=32$ 的实指数衰减序列, 只保留部分系数, 比较 DFT 和 DCT 的能量紧致性。

给出一个最简单的保留系数序号的选择, 对于 DFT, 若只保留一个系数, 就保留序号 $k=0$ 的系数(直流分量), 若保留更多系数, 因为实序列 DFT 的系数满足共轭对称性 $X[k] = X^*[N-k]$, 保留的非零系数必须保持该性质, 因此序号 $k, N-k$ 必须同时保留。例如, 若保留 5 个系数, 则保留系数序号集为 $\Pi = \{0, 1, 2, N-2, N-1\}$, 对于 DCT 系数, 问题简单得多, 若保留 M 个系数, 只需保留 $0 \leq k \leq M-1$ 序号的系数, 若保留 5 个 DCT 系数, 则 $\Pi_{\text{DCT}} = \{0, 1, 2, 3, 4\}$ 。

若保留系数个数为 $|\Pi| = |\Pi_{\text{DCT}}| = N$ 时, 全部系数被保留, 反变换重构的信号无失真, 当 $|\Pi| = |\Pi_{\text{DCT}}| = 1$ 时, 只保留一个直流系数, 重构信号是常数, 误差较大。改变保留系数数目, 计算重构误差, 结果示于图 3.32 中。注意图中横坐标是被置为零的系数个数, 即 $N - |\Pi|$, 纵坐标是均方误差, 即式(3.76)和式(3.77)误差项除以 N 得到的平均误差。可见, 在此例中, DCT 的紧致性比 DFT 好得多。

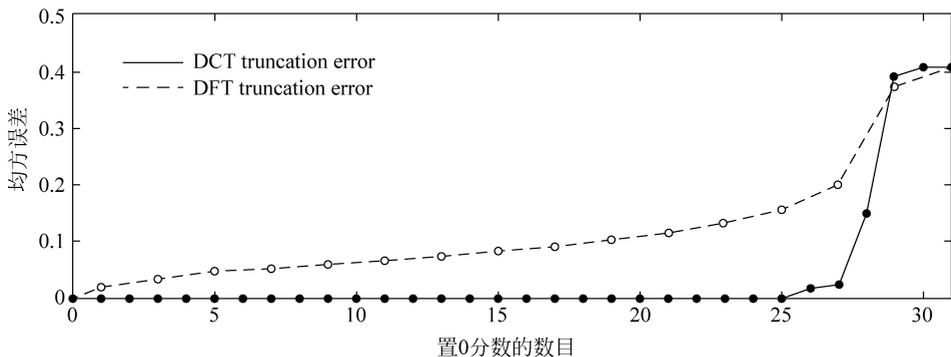


图 3.32 DCT 和 DFT 能量紧致性的比较

注意到, 能量紧致性不是从数学上严谨地证明的特性, 而是一种统计结果。我们说 DCT 的紧致性比 DFT 好, 意味着对大多数实际信号讲, 这是成立的; 但也能找到一些信号, DFT 的紧致性可能更好。例如, 对于信号 $A \cos\left(\frac{2\pi}{N}mn\right) + B \cos\left(\frac{2\pi}{N}kn\right)$, 这里 m 和 k 都是整数, 对这种特殊信号, DFT 的紧致性是最好的。

3.8.2 离散余弦变换的快速算法

由于 DCT 获得广泛应用, DCT 的快速算法得到深入的研究。这里简单介绍两类 DCT 的快速算法。第一类是基于 FFT 的间接算法, 第二类是针对 DCT 直接导出的快速算法。

基于 FFT 的快速算法, 利用 DCT 与一些特殊序列 DFT 的关系, 直接利用 FFT 计算 DFT 系数后, 得到 DCT 系数。显然, 第一个算法是利用式(3.66)得到 $2N$ 点对称序列, 用 FFT 得到 $2N$ 点 DFT 系数, 再通过式(3.68)得到 DCT 系数。若做 DCT 反变换, 由 N 点 DCT 系数通过式(3.69)得到 $2N$ 点 DFT 系数通过 IFFT 得到 $2N$ 点序列 $x_2[n]$, 再取出前 N 个点。

一种稍简单的方法是定义 $2N$ 点序列

$$x_1[n] = \begin{cases} x[n], & 0 \leq n < N \\ 0, & N \leq n < 2N - 1 \end{cases}$$

对 $x_1[n]$ 做 $2N$ 点 DFT 得到 $X_1[k]$, 可以证明(留作习题)

$$X_{\text{DCT}}[k] = \frac{1}{\sqrt{N}} \text{Re} \{ e^{-j\frac{\pi}{2N}k} X_1[k] \}, \quad 0 \leq k < N$$

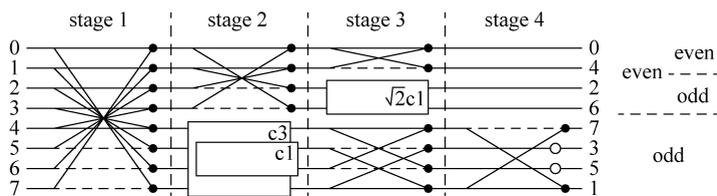
这里 $\text{Re}\{\cdot\}$ 指取实部。

DCT 自身存在高效的快速算法, 利用 DCT 的性质可构造专用的快速算法。这方面结果很多, 进一步可以参考文献[59]。这里仅简要介绍 Loeffler 等提出的一个快速算法, 它的结构规范, 适用于硬件实现。对于 8 点 DCT 仅需要 11 次乘法和 29 次加法, 乘法次数达到理论上的下限。这个算法的流程图如图 3.33(a) 所示, 其中图 3.33(a) 中的主要运算单元如图 3.33(b) 所示。注意, 在图(b)的运算单元中, 用下列等式可以将 4 次乘法、2 次加法运算变成 3 次乘法和 3 次加法

$$\begin{cases} y_0 = ax_0 + bx_1 = (b-a)x_1 + a(x_0 + x_1) \\ y_1 = -bx_0 + ax_1 = -(a+b)x_0 + a(x_0 + x_1) \end{cases} \quad (3.78)$$

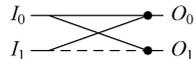
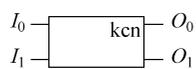
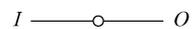
由于 a, b 是常数, $a+b, b-a$ 可以预先算好存在寄存器中, 不需要增加运算。对于快速反变换, 只需要改换输入/输出方向。

在图像压缩标准 JPEG 或视频压缩标准 MPEG 中, 对 8×8 的块图像数据进行 DCT 变换, 可通过多次对 8 点序列的 DCT 完成, 因此 8 点 DCT 是实际中最常用的。图 3.33 的算法流程图很有用。



(a) 8点DCT的计算流程

图 3.33 8 点 DCT 的快速算法

symbol	equations	effort
	$O_0 = I_0 + I_1$ $O_1 = I_0 - I_1$	2 add
	$O_0 = I_0 \cdot k \cdot \cos \frac{\pi k}{2N} + I_1 \cdot k \cdot \sin \frac{\pi k}{2N}$ $O_1 = -I_0 \cdot k \cdot \sin \frac{\pi k}{2N} + I_1 \cdot k \cdot \cos \frac{\pi k}{2N}$	3 mult+ 3 add
	$O = \sqrt{2} \cdot I$	1 mult

(b) 计算流程中的运算单元

图 3.33 (续)

* 3.9 一些其他离散变换简介

除 DFT 和 DCT 外,信号处理中还用到若干离散正交变换,这些变换在不同领域得到了应用,本节简述几种其他的离散变换。

3.9.1 离散正弦变换

与离散余弦变换对应,存在一种离散正弦变换(DST),一维 DST 变换的基函数集是

$$a_k[n] = \sqrt{\frac{2}{N+1}} \sin \frac{\pi(n+1)(k+1)}{N}, \quad 0 \leq k \leq N-1, 0 \leq n \leq N-1 \quad (3.79)$$

因此,矢量 $\mathbf{x} = [x[0], x[1], \dots, x[N-1]]^T$ 的 DST 定义为

$$X_{\text{DST}}[k] = \sqrt{\frac{2}{N+1}} \sum_{n=0}^{N-1} x[n] \sin \frac{\pi(k+1)(n+1)}{N+1}, \quad 0 \leq k \leq N-1 \quad (3.80)$$

反变换(IDST)为

$$x[n] = \sqrt{\frac{2}{N+1}} \sum_{k=0}^{N-1} X_{\text{DST}}[k] \sin \frac{\pi(k+1)(n+1)}{N+1}, \quad 0 \leq n \leq N-1 \quad (3.81)$$

DST 可以通过对序列进行反对称延拓,建立起与 DFT 的联系。本质上,DST 与 DCT 一样,也是 KL 变换的一种逼近。

有学者进一步定义了离散余弦变换和离散正弦变换族,包括四类余弦变换和四类正弦变换。其中 3.8 节讨论的也是最常用的这种离散余弦变换称为第二类余弦变换(DCT-II)。关于余弦和正弦变换族的进一步讨论,请参考文献[89]。

3.9.2 Hadamard 变换

除一个归一化因子外,Hadamard 变换的基序列仅由 1、-1 组成,适宜于数字信号的合成与分解。

设变换序列 $N=2^n$,Hadamard 变换矩阵 \mathbf{H}_n 写成如下递推关系

$$\mathbf{H}_n = \frac{1}{\sqrt{2}} \begin{pmatrix} \mathbf{H}_{n-1} & \mathbf{H}_{n-1} \\ \mathbf{H}_{n-1} & -\mathbf{H}_{n-1} \end{pmatrix}$$

由初始值

$$\mathbf{H}_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

递推可以得到长度为 8 的变换矩阵为

$$\mathbf{H}_3 = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

3.9.3 Haar 变换

Haar 变换相当于用两个滤波器 $H_1 = \left\{ \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right\}$ 和 $H_h = \left\{ \frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}} \right\}$, 分别对数据序列滤波, 并进行 2:1 降采样。得到的高频输出部分保持, 而低频部分继续这样的分解过程, 直到最后只有一个点。设 $N=2^n$, 分解过程需进行 n 次, 这个过程也等价为一个线性变换。 $N=8$ 的 Haar 变换矩阵为

$$\mathbf{H}_{r3} = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$

第 8 章将会看到, Haar 变换是一种最简单的小波变换。

3.9.4 Slant 变换

Slant 变换的核函数也可以通过递推公式求出, 此处从略, 仅给出一个 $N=2^2$ 的变换矩阵的例子

$$\mathbf{S}_2 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} \\ 1 & -1 & -1 & 1 \\ \frac{1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & \frac{-1}{\sqrt{5}} \end{bmatrix}$$

还有很多通过特别方法构造的正交变换,这些变换各有一些特殊性质,并获得一些应用。正交变换很多[58,59],目前非正交的变换甚至突破基函数概念的框架变换、词典变换等的研究也很活跃,有兴趣的读者可参考有关文献[30]。

3.10 与本章相关的 MATLAB 函数与实例

3.10.1 相关的 MATLAB 函数简介

1. fft

功能介绍 离散傅里叶变换(FFT)。

语法

```
y = fft(x)
y = fft(x, n)
y = fft(x, n, dim)
```

输入变量 $y = \text{fft}(x)$ 利用 FFT 算法计算矢量 x 的离散傅里叶变换。 x 表示原始信号,当 x 的长度小于 n 时,在 x 的尾部补零,构成 n 点数据;当 x 的长度大于 n 时, x 被截断成为 n 点数据。若 x 为矩阵,则 fft 函数作用于 x 的每一列。 dim 表示 fft 函数作用于 x 的维度。

输出内容 y 代表输出信号。

2. ifft

功能介绍 离散傅里叶反变换(IFFT)。

语法

```
y = ifft(x)
y = ifft(x, n)
y = ifft(x, n, dim)
```

输入变量 $y = \text{ifft}(x)$ 利用 IFFT 算法计算矢量 x 的离散傅里叶反变换。 x 表示原始变换系数,当 x 的长度小于 n 时,在 x 的尾部补零,构成 n 点数据;当 x 的长度大于 n 时, x 被截断成为 n 点数据。若 x 为矩阵,则 ifft 函数作用于 x 的每一列。 dim 表示 ifft 函数作用于 x 的维度。

输出内容 y 代表输出信号。

3. fftshift

功能介绍 对 fft 的输出重新排列,将零频分量移到频谱的中心。

语法

```
y = fftshift(x)
```

输入变量 $y = \text{fftshift}(x)$ 对 x 重新排列。 x 表示原始变换系数,当 x 为向量时,将 x 的左右部分交换;当 x 为矩阵时,将 x 的第 1,3 象限以及 2,4 象限分别交换。

输出内容 y 代表输出信号。

4. czt

功能介绍 线性调频 z 变换。

语法

```
y = czt(x, m, w, a)
```

输入变量 $y = \text{czt}(x, m, w, a)$ 计算 x 的线性调频 z 变换。 x 表示原始信号, m 为要分析的频谱点数, w 为 z 平面取样轮廓线上各点之间的比率, a 为轮廓线上的复数起点。

输出内容 y 代表输出信号。

5. dct

功能介绍 离散余弦变换。

语法

```
y = dct(x)
y = dct(x, n)
```

输入变量 $y = \text{dct}(x)$ 计算矢量 x 的离散余弦变换。 x 表示原始信号, 当 x 的长度小于 n 时, 在 x 的尾部补零, 构成 n 点数据; 当 x 的长度大于 n 时, x 被截断成为 n 点数据。若 x 为矩阵, 则 dct 函数作用于 x 的每一列。

输出内容 y 代表输出的变换系数。

6. idct

功能介绍 离散余弦反变换。

语法

```
y = idct(x)
y = idct(x, n)
```

输入变量 $y = \text{idct}(x)$ 计算矢量 x 的离散余弦反变换。 x 表示原始变换系数, 当 x 的长度小于 n 时, 在 x 的尾部补零, 构成 n 点数据; 当 x 的长度大于 n 时, x 被截断成为 n 点数据。若 x 为矩阵, 则 idct 函数作用于 x 的每一列。

输出内容 y 代表输出信号。

3.10.2 MATLAB 例程

例 3.10.1 已知信号 $x[n] = [0, 1, 2, 3, 4, 5, 6, 7]$, 计算其离散傅里叶变换 $X[k]$, 中心为零频的序列 $\hat{X}[k]$, 离散傅里叶反变换 $\hat{x}[n]$, 并画出相应的图形。

```
x = [0 1 2 3 4 5 6 7];
y = fft(x);
y1 = fftshift(y);
x1 = ifft(y);
figure(1);
stem(x, 'ko');
xlabel('samples');
```

```

ylabel('Amplitude');
figure(2);
stem(abs(y), 'ko');
xlabel('samples');
ylabel('Amplitude');
figure(3);
stem(abs(y1), 'ko');
xlabel('samples');
ylabel('Amplitude');
figure(4);
stem(x1, 'ko');
xlabel('samples');
ylabel('Amplitude');

```

运行结果如图 3.34 所示。

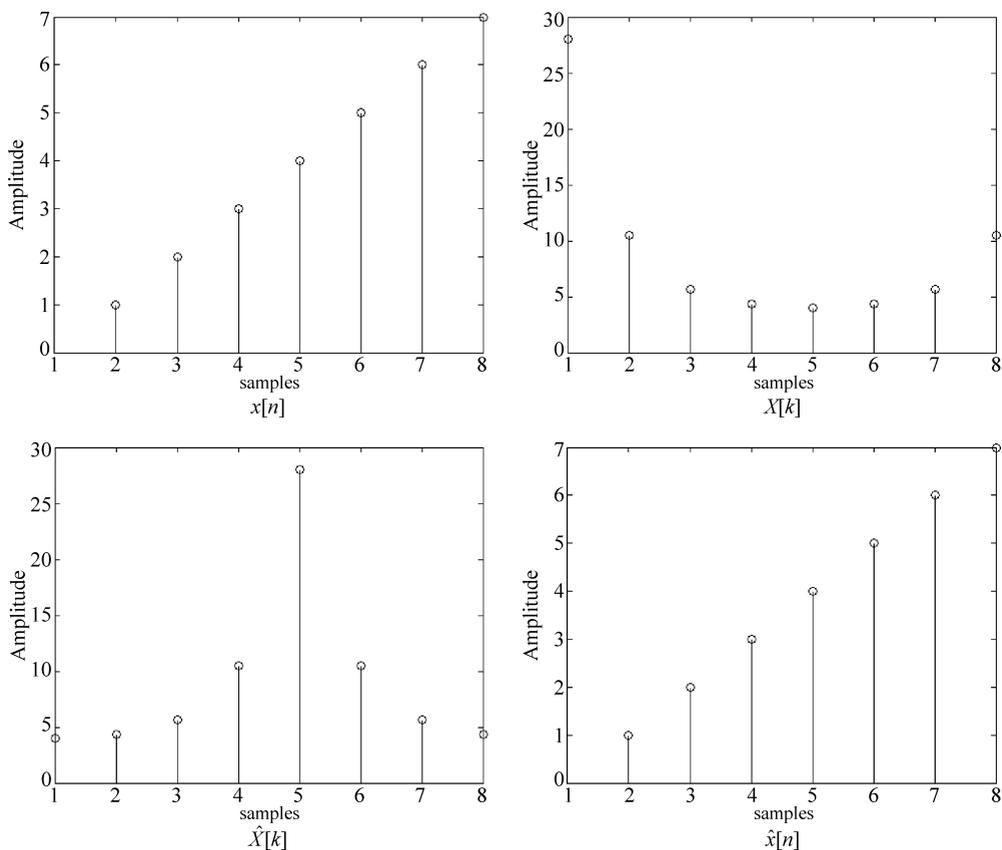


图 3.34 例 3.10.1 的例程结果

例 3.10.2 设 $x[n]$ 是由 3 个实正弦信号组成, 频率分别为 7Hz、8Hz、9Hz, 采样频率为 50Hz, 采样点数为 256, 在 6~10Hz 频率段求其 CZT。

```

t = 0:1/50:255/50;
x = sin(2 * pi * 7 * t) + sin(2 * pi * 8 * t) + sin(2 * pi * 9 * t);
fs = 50;
f1 = 6;
f2 = 10;
m = 50;
w = exp(-j * 2 * pi * (f2 - f1)/(m * fs));
a = exp(j * 2 * pi * f1/fs);
y = czft(x, m, w, a);
fy = (0:(m - 1)) * (f2 - f1)/m + f1;
plot(fy, abs(y), 'ko');
title('CZT');
xlabel('Frequency');
ylabel('Amplitude');

```

运行结果如图 3.35 所示。

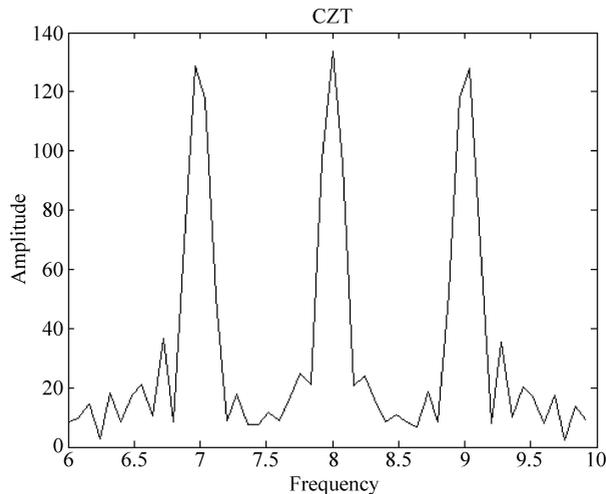


图 3.35 例 3.10.2 的例程结果

例 3.10.3 计算信号 $x[n] = 2n + 100\cos(2\pi n/5)$, $n = 1, 2, \dots, 50$ 的 DCT 和 IDCT。

```

n = 1:50;
x = 2 * n + 100 * cos(2 * pi * n/5);
y = dct(x);
x1 = idct(y);
figure(1);
stem(abs(y), 'ko');
xlabel('samples');
ylabel('Amplitude');

figure(2);
stem(x1, 'ko');
xlabel('samples');
ylabel('Amplitude');

```

运行结果如图 3.36 所示。

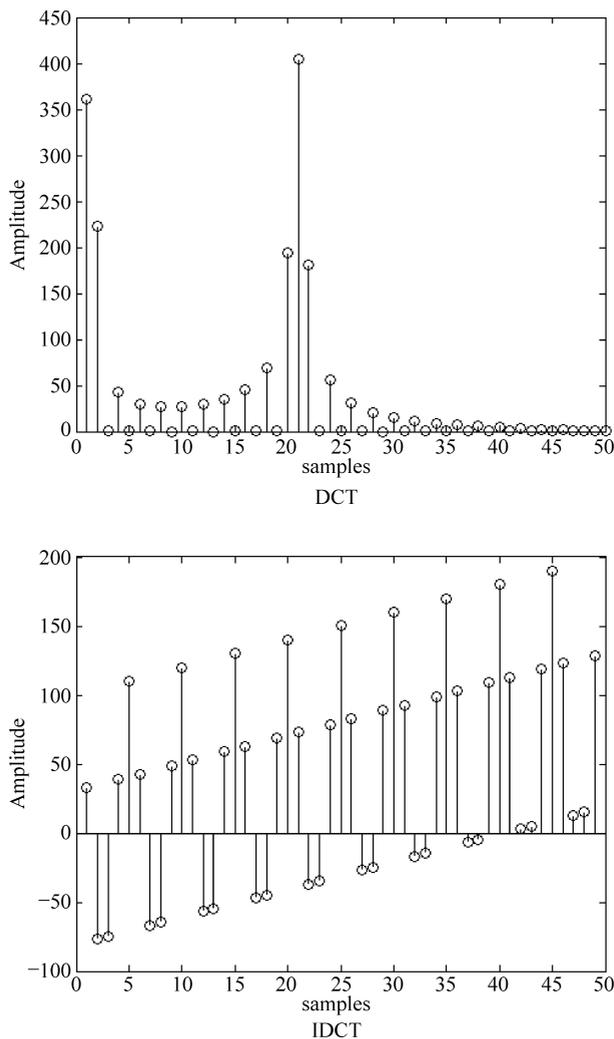


图 3.36 例 3.10.3 的例程结果

3.11 本章小结

DFT 和 DCT 等变换都是基于正交基的正交变换。正交变换的一个明显优势是正、反变换计算上的简洁性。在很多应用中,人们首先选择正交基,正交变换是信号表示中的一种最基本的变换方式。随着研究的深入,也出现很多非正交基,甚至基于“非基”序列集的变换。例如利用更广义的“框架”替代基序列集,构成的一般变换,在表示信号的稀疏性和时频局域性等方面显示出优势。本书主要讨论有关正交变换及其应用,属于“框架”变换的离散小波变换将在第 8 章给出简要介绍,有关信号变换的更广泛的讨论,可参考有关文献,例如 Mallat 的文献[30]。

习题

3.1 计算如下信号的 N 点 DFT (N 为偶数)。

$$(1) x[n] = R_N[n] e^{-j\frac{2\pi}{N}n}$$

$$(2) x[n] = \delta[n - n_1] - \delta[n - n_2], 0 < n_1, n_2 < N$$

$$(3) x[n] = \begin{cases} 1, & n \text{ 为偶数}, 0 \leq n < N \\ 0, & n \text{ 为奇数}, 0 \leq n < N \end{cases}$$

$$(4) x[n] = \begin{cases} 1, & 0 \leq n < N/2 \\ 0, & N/2 \leq n < N-1 \end{cases}$$

$$3.2 \text{ 已知信号 } x[n] = \begin{cases} \cos\left(\frac{\pi n}{6}\right), & 0 \leq n < 12 \\ 0, & \text{其他} \end{cases}。$$

(1) 求 $N=12$ 点的 DFT $X[k]$;

(2) 若取 $x[n]$ 的前 24 点, 求 $N=24$ 点的 DFT $X_1[k]$ 。

3.3 如果 $\tilde{x}[n]$ 是周期为 N 的周期序列, 那么它也是周期为 $2N$ 的周期序列。如果把 $\tilde{x}[n]$ 看作周期为 N 的周期序列, 令 $\tilde{X}_1[k]$ 表示 $\tilde{x}[n]$ 的离散傅里叶级数的系数; 当把 $\tilde{x}[n]$ 看作周期为 $2N$ 的周期序列, 令对应的傅里叶级数的系数为 $\tilde{X}_2[k]$ 。请利用 $\tilde{X}_1[k]$ 来确定 $\tilde{X}_2[k]$ 。

3.4 若 $X[k] = \text{DFT}\{x[n]\}$, 试证明下列各式。

$$(1) \text{DFT}\{x[n]\} = Nx[N-k]$$

$$(2) X[0] = \sum_{n=0}^{N-1} x[n]$$

$$(3) x[0] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]$$

$$(4) \text{若 } N \text{ 为偶数, 则 } X\left[\frac{N}{2}\right] = \sum_{n=0}^{N-1} (-1)^n x[n]$$

3.5 已知连续时间信号 $x_a(t) = e^{j(\sqrt{2}\pi t + \varphi_0)}$ 为复单频信号, 现以 T 为周期对它进行采样得到离散序列 $x[n] = e^{j(\sqrt{2}\pi nT + \varphi_0)}$ 。对该序列做 N 点 DFT 得到 N 个变换系数。如果要使得 DFT 的 N 个变换系数只有一个不为零而其他全部为零, T 和 N 应满足什么条件? 为什么?

3.6 已知 N 点序列 $x[n]$ 的 N 点 DFT 是 $X[k]$ 。设 $y[n]$ 定义如下:

$$y[n] = \begin{cases} x[n], & 0 \leq n < N \\ 0, & N \leq n < 3N \end{cases}$$

对 $y[n]$ 做 $3N$ 点的 DFT, 请用 $X[k]$ 表示 $Y[k]$ 。

3.7 已知 N 点序列 $x[n]$ 的 N 点 DFT 是 $X[k]$ 。对如下定义的 N 点序列求 N 点 DFT, 请用 $X[k]$ 表示新的变换系数:

$$(1) y_1[n] = x[N-1-n], 0 \leq n < N$$

$$(2) y_2[n] = (-1)^n x[n], 0 \leq n < N$$

3.8 已知 N 点序列 $x[n]$ 的 N 点 DFT 是 $X[k]$ 。对如下定义的 $2N$ 点序列求 $2N$ 点 DFT, 请用 $X[k]$ 表示新的变换系数:

$$(1) y_1[n] = \begin{cases} x[n], & 0 \leq n < N \\ x[n-N], & N \leq n < 2N \\ 0, & \text{其他} \end{cases}$$

$$(2) y_2[n] = \begin{cases} x[n], & 0 \leq n < N \\ 0, & N \leq n < 2N \\ 0, & \text{其他} \end{cases}$$

$$(3) y_3[n] = \begin{cases} x[n/2], & n \text{ 为偶数}, 0 \leq n < 2N \\ 0, & n \text{ 为奇数}, 0 \leq n < 2N \end{cases}$$

3.9 一个长度为 N 的有限长序列, 从 0 时刻起有连续 N 个非零值, 该序列的 z 变换为 $X(z)$ 。对该序列补 $2N$ 个零, 做 $3N$ 点长度的 DFT, 用 $X(\cdot)$ 表示第 $k=5$ 的 DFT 系数值。

3.10 已知 N 点序列 $x[n]$ 的 N 点 DFT 是 $X[k]$, 请证明:

(1) 如果 $x[n]$ 是实周期偶对称, 即 $x[n] = x[N-n]$, 则 $X[k]$ 也具有实偶对称性。

(2) 如果 $x[n]$ 是实周期奇对称, 即 $x[n] = -x[N-n]$, 则 $X[k]$ 为纯虚数且奇对称。

3.11 已知 N 点序列 $x[n]$ 的 N 点 DFT 是 $X[k]$ 。可通过如下三种方式将其扩展成 rN 点的序列。

(1) 在原序列后面补零, 即

$$y_1[n] = \begin{cases} x[n], & 0 \leq n < N \\ 0, & N \leq n < rN \end{cases}$$

(2) 在原序列两点间插入 $r-1$ 个零, 即

$$y_2[n] = \begin{cases} x[n/r], & n = mr, \quad 0 \leq m < N \\ 0, & n \neq mr, \quad 0 \leq m < N \end{cases}$$

(3) 将原序列进行 r 次的周期延拓, 即

$$y_3[n] = x[(n)_N] R_{rN}[n]$$

求上述三个 rN 点序列的 rN 点 DFT $Y_1[k]$ 、 $Y_2[k]$ 和 $Y_3[k]$ 与 $X[k]$ 之间的关系。

3.12 设 $X[k]$ 是一个 8 点长实序列 $x[n]$ 的 8 点 DFT, 已知其中 5 个值 $X[0] = X[1] = 0, X[2] = 4, X[3] = X[4] = 0$, 求 $\sum_{n=0}^7 x^2[n]$ 。

3.13 一个 4 点序列的 DFT 系数分别为: $\{2, 1+j, 0, 1-j\}$, 不加运算能否判断原序列是实序列还是复序列, 为什么? 序列的零序号值 $x[0]$ 为多少?

3.14 已知长度为 9 的序列 $x[n] = \{-2, 3, 1, 4, -3, 0, 2, -1, 6\}$, 其 DTFT 为 $X(e^{j\omega})$ 。

(1) 从 $\omega=0$ 开始在 $[0, 2\pi)$ 区间以间隔 $\pi/6$ 对 $X(e^{j\omega})$ 进行均匀采样, 得到序列 $X_1[k]$ 。 $x_1[n]$ 为 $X_1[k]$ 对应的 IDFT 序列, 不计算 $X(e^{j\omega})$ 和 $X_1[k]$ 请直接写出序列 $x_1[n]$ 。

(2) 从 $\omega=0$ 开始在 $[0, 2\pi)$ 区间以间隔 $\pi/4$ 对 $X(e^{j\omega})$ 进行均匀采样, 得到序列 $X_2[k]$ 。 $x_2[n]$ 为 $X_2[k]$ 对应的 IDFT 序列, 不计算 $X(e^{j\omega})$ 和 $X_2[k]$ 请直接写出序列 $x_2[n]$ 。

3.15 已知无限长序列 $x[n]$ 的 z 变换为 $X(z)$, 且其收敛域包含单位圆; 另一长度为 N 的序列 $y[n]$ 的 N 点 DFT 为 $Y[k]$ 。已知 $Y[k]$ 和 $X(z)$ 间满足如下关系

$$Y[k] = X(z) \Big|_{z=e^{j\frac{2\pi}{N}k}}, \quad 0 \leq k < N$$

求 $y[n]$ 和 $x[n]$ 间的关系。

3.16 已知 5 点序列 $x[n] = 2\delta[n] + \delta[n-1] - \delta[n-4]$ 。

(1) 求对应的 DFT $X[k]$;

(2) 分别求 $x_1[n] = \text{IDFT}\{X^2[k]\}$ 和 $x_2[n] = \text{IDFT}\{|X[k]|^2\}$, 并说明这两个结果的物理意义。

3.17 已知长度为 12 的实序列 $x[n]$ 的 12 点 DFT 为 $X[k]$, $X[k]$ 的前 7 个样本取值为 $X[k] = \{12, -18 - 21j, -10 + 4j, -6 + 7j, 9 + 8j, 19 - 16j, 39\}$, $0 \leq k \leq 6$ 。不计算 $X[k]$ 的 IDFT, 其如下表达式的值。

$$(1) x[0] \quad (2) x[6] \quad (3) \sum_{n=0}^{11} x[n] \quad (4) \sum_{n=0}^{11} |x[n]|^2 \quad (5) \sum_{n=0}^{11} e^{j\pi n/3} x[n]$$

3.18 已知 12 点序列 $x[n] = \{2, -3, 4, -4, 5, -3, 3, 0, -1, -1, -1, 1\}$, 其离散时间傅里叶变换为 $X(e^{j\omega})$ 。要求通过 M 点 DFT 来计算 $X(e^{j\frac{8}{3}\pi})$ 。

(1) 确定 M 可取的最小值;

(2) 按照最小 M 取值给出通过 DFT 计算 $X(e^{j\frac{8}{3}\pi})$ 的过程和结果。

3.19 已知如题 3.19 图所示的 5 点长序列, 画出下列各序列的图形

$$(1) \tilde{x}_1[n] = x[((-n))_5]$$

$$(2) \tilde{x}_2[n] = x[((-n))_3]$$

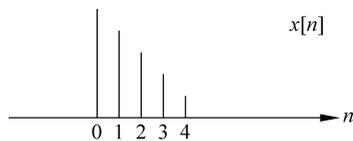
$$(3) \tilde{x}_3[n] = x[((-n))_7]$$

$$(4) x_4[n] = x[((n-2))_5]R_5[n]$$

$$(5) x_5[n] = x[((2-n))_3]R_5[n]$$

$$(6) x_6[n] = x[((n-2))_7]R_7[n]$$

$$(7) x_{ep}[n] \text{ 和 } x_{op}[n]$$



题 3.19 图

3.20 已知两个序列分别为

$$x_1[n] = \delta[n] + \delta[n-1] - \delta[n-2] - \delta[n-3]$$

$$x_2[n] = \delta[n] - \delta[n-2] + \delta[n-4]$$

(1) 求两个序列的线性卷积 $y_1[n] = x_1[n] * x_2[n]$;

(2) 求两个序列的 5 点循环卷积 $y_2[n] = x_1[n] \otimes x_2[n]$, $N=5$;

(3) 求两个序列的 8 点循环卷积 $y_3[n] = x_1[n] \otimes x_2[n]$, $N=8$ 。

3.21 已知 $x_1[n]$ 是 100 点长的序列, 其非零值范围为 $0 \leq n \leq 99$; $x_2[n]$ 是 20 点长的序列, 其非零值范围为 $10 \leq n \leq 39$ 。两序列做 100 点的循环卷积得到 $y[n]$ 。请确定 $y[n]$ 中哪些 n 值对应 $x_1[n]$ 和 $x_2[n]$ 线性卷积的结果。

3.22 计算一个长度为 5000 点的序列与一个长度为 100 点序列的线性卷积, 要求利用重叠相加法并通过 256 点的 FFT 和 IFFT 来实现。请回答至少需要多少次 FFT 和多少次 IFFT, 并请详细说明理由。

3.23 已知 $x[n]$ 是 N (N 为偶数) 点的有限长序列, 它的 DFT 是 $X[k]$ 。

(1) 如果由 $X[k]$ 中所有奇次谐波组合成新的 $\frac{N}{2}$ 点频谱 $X_1[r]$ (即 $X_1[r]=X[2r+1]$, $0 \leq r \leq \frac{N}{2}-1$), 试推导 $x_1[n]=\text{IDFT}\{X_1[r]\}$ ($0 \leq n \leq \frac{N}{2}-1$) 与原序列 $x[n]$ 的关系?

(2) 根据以上的推导, 问对 $x[n]$ 经过怎样的预处理, 就可以用 $\frac{N}{2}$ 点的复数 FFT 模块得到 $X[k]$ 中全部的奇次谐波频谱, 以 $N=8$ 为例画出处理的流程框图。

3.24 在按时间抽取基 2 的 128 点 FFT 算法中, 第一级中与 $x[47]$ 组成同一个蝶形的是 $x[m]$, 请确定 m 的取值。

3.25 对一个长度为 $N=2^m$ 的序列 $x[n]$ 做 N 点 DFT。如果只需要计算 DFT 的奇数序号值, 即 $X[1], X[3], \dots, X[N-1]$, 能否用一个 $\frac{N}{2}$ 点的 DFT 处理器进行计算? 如果可以, 请写出 $\frac{N}{2}$ 点 DFT 处理器的输入序列的一般表达式。

3.26 设有一个 $N=2^m$ 点的复数基 2 FFT 运算模块, 要求利用它一次算得 $2N$ 点的实序列的 DFT。请说明计算方法, 并画出相应的处理框图。

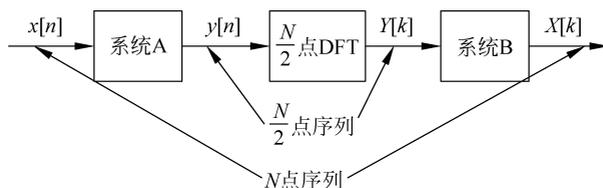
3.27 序列 $x[n]$ 的长度为 N , 其中 N 为偶数。 $x[n]$ 的 N 点 DFT 可以按如下方式计算: 将 $x[n]$ 拆成两个 $\frac{N}{2}$ 点序列 $g_1[n]=x[2n]$ 和 $g_2[n]=x[2n+1]$, 分别计算这两个序列 $\frac{N}{2}$ 点 DFT 得到 $G_1[k]$ 和 $G_2[k]$, 然后将其进行组合可得到 $X[k]$ 。如果在构建 $g_1[n]$ 和 $g_2[n]$ 时出现了错误, 使得 $g_1[n]=x[2n+1]$ 、 $g_2[n]=x[2n]$, 但仍按原来方式进行计算, 这样就会得到了一个错误的结果 $\hat{X}[k]$ 。请利用 $X[k]$ 给出 $\hat{X}[k]$ 的表达式。

3.28 设 $x[n]$ 是长度为 N 的序列, 其中 N 是偶数, 且

$$x[n] = -x\left[n + \frac{N}{2}\right], \quad n = 0, 1, \dots, \frac{N}{2} - 1$$

(1) 证明: 在 $x[n]$ 的 N 点 DFT $X[k]$ 中, 当 k 为偶数时, $X[k]=0$;

(2) 能否用如题 3.28 图所示的一个 $\frac{N}{2}$ 点的 DFT 处理器, 通过级联系统 A 和系统 B 进行计算? 图中 $y[n]$ 和 $Y[k]$ 长度均为 $\frac{N}{2}$ 点, $Y[k]$ 是 $y[n]$ 的 $\frac{N}{2}$ 点 DFT。如果可以, 请推导出 $x[n]$ 与 $y[n]$ 的关系, 以及 $X[k]$ 与 $Y[k]$ 的关系。



题 3.28 图

3.29 一个序列长为 $N=16$,按频率抽取进行基 4 分解计算其 FFT。

(1) 请写出第一级分解的原理表达式;

(2) 画出第一级自上而下的第 2 个蝶形和第二级分解的自上而下第 3 个蝶形的蝶形图(注意,蝶形图画标准的基 4 蝶形,不需要将基 4 蝶形分解成基 2 蝶形,标注蝶形的输入、输出及各支路系数)。

3.30 对于长度 $N=64$ 的序列,进行基 4 按时间抽取 FFT,按从上到下,写出第二级分解对应的前 8 个旋转因子。

3.31 一个长度为 9 的序列,按 3×3 的时间抽取进行 FFT,推导它的按时间抽取方式分解的原理表达式,画出完整的蝶形图,分析其乘法运算量(其中,乘以 $\pm 1, \pm j$ 不计作一次乘法),比较它与直接补零用 16 点分裂基的运算量(使用分裂基算法时,不考虑一些输入信号是否是由补零产生)。

3.32 在一些实际应用中,一边采集信号一边进行 DFT 运算,当信号采集到 N 个样本后,得到信号序列

$$\mathbf{x}^{(1)} = \{x[0], x[1], \dots, x[N-2], x[N-1]\}$$

用 FFT 进行计算,得到这第一组 DFT 系数 $X^{(1)}[k]$ 。若信号采集继续进行,信号存于固定长度的先入先出的存储器中,在继续采集了 M 个样本后,存储器里存的信号是

$$\mathbf{x}^{(2)} = \{x[M], x[M+1], \dots, x[N+M-2], x[N+M-1]\}$$

对这一组信号的 DFT 记为 $X^{(2)}[k]$,这样的过程,称为滑动 DFT 或离散短时傅里叶变换。常用两类方法计算滑动 DFT,一种是对每一组信号都直接采用 FFT,一种计算方法是通过对已计算出的 $X^{(1)}[k]$ 计算 $X^{(2)}[k]$,

(1) 导出由 $X^{(1)}[k]$ 和一些采样值联合计算 $X^{(2)}[k]$ 的一般表达式;

(2) 假设 $M=2^m, M < N=2^l$,对如上导出的各组运算尽可能使用 FFT,估计其运算量,并与直接进行 FFT 的算法进行比较。

3.33 证明 DCT 系数满足对称关系: $X_{\text{DCT}}[k] = -X_{\text{DCT}}[2N-k], 0 \leq k < 2N$ 。

3.34 定义 $2N$ 序列点序列

$$x_1[n] = \begin{cases} x[n], & 0 \leq n < N \\ 0, & N \leq n < 2N-1 \end{cases}$$

对 $x_1[n]$ 做 $2N$ 点 DFT 得到 $X_1[k]$,证明 DCT 变换可由下式求得

$$X_{\text{DCT}}[k] = \partial[k] \frac{1}{\sqrt{N}} \text{Re}\{e^{-j\frac{\pi}{2}k} X_1[k]\}, 0 \leq k < N$$

这里 $\text{Re}\{\cdot\}$ 指取实部。

3.35 证明式(3.76)和式(3.77)。

MATLAB 习题

3.1 请用 MATLAB 画出习题 3.2 中序列 $x[n]$ 的 DTFT,然后利用 MATLAB 自带 fft 函数计算序列 $x[n]$ 的 12 点和 24 点的 DFT,并将计算结果与 DTFT 画在同一张图中。通过该练习理解“DFT 作为对 DTFT 的频域离散采样”的含义。

3.2 请利用 MATLAB 自带 fft 函数计算习题 3.20 中两个序列的线性卷积和循环卷

积,并与按照定义直接计算的结果进行比对。

3.3 请利用 MATLAB 编写任意 2 的整数次幂点数的基 2 FFT 的通用函数,并与 MATLAB 自带 `fft` 函数进行比对确认自己编写函数的正确性。

3.4 请利用 MATLAB 按照 3.7 节中介绍的用 FFT 计算 CZT 的算法编写计算 CZT 的通用函数,并与 MATLAB 自带的 CZT 函数进行比对确认自己编写函数的正确性。

3.5 请利用 MATLAB 验证 3.4 节所讨论的 DFT 各种性质。

频谱分析是信号分析中的核心问题之一,有非常广泛的应用。频谱分析是分析信号物理特性的有效手段。谱分析历史上可追溯到牛顿时代,牛顿的光谱试验开启了谱分析的先河,而信号处理的频谱分析至今仍是活跃的研究方向。

从信号处理角度讲,建立在傅里叶变换基础上的频谱概念,因其物理意义明确,成为信号频谱分析的基础。在数字信号处理被广泛应用之前,基于傅里叶变换的频谱分析的概念早已被人们所认识;随着 DFT/FFT 计算技术的发展,基于 DFT 的数字频谱分析成为应用最广泛的工具。一方面,DFT 频谱分析工具应用得越来越广泛;另一方面,人们也认识到 DFT 频谱分析的局限性。因此,许多新的技术被提出,例如高分辨谱分析技术。同时,由于数字信号合成技术被广泛采用,更多的复杂信号被使用,很多复杂信号具有典型的时变谱特性,因此人们又提出了时频谱分析的方法。时频谱进一步拓展了人们关于信号稀疏性的认识,刺激了利用稀疏性有效处理信号的一些新技术的发展。从这个简短的叙述可以看出,谱分析既是信号处理中最经典的内容,又与前沿研究紧密相关,是信号处理的重要课题。

本章概要介绍基于 DFT 的频谱分析技术,这是频谱分析的基础。一方面,通过 DFT 谱分析透彻理解频谱分析的一些重要概念,为学习更高级的谱分析技术打下基础;另一方面,DFT 谱分析仍是工程中使用最多的方法。在 DFT 谱分析基础上推广到短时傅里叶变换,初步学习有关时频分析的概念。

4.1 DFT 与连续信号频谱的关系

DFT 作为频谱分析的主要工具,理解 DFT 的每一个系数与对应的连续信号频谱之间的关系是有意义的。当离散信号是由连续信号采样获得时,本节以两种思路讨论 DFT 系数与连续信号频谱之间的关系。第一种方法比较直观,利用 DFT 作为 DTFT 的采样值和 DTFT 与连续信号频谱的直接对应关系,得到 DFT 系数物理意义的一种直观解释;第二种方法,从 DFT 的定义、离散采样和连续信号频谱定义出发,通过演绎过程直接找到连续信号频谱与离散信号频谱的一般关系,利用导出的结果可以从滤波器的观点进一步理解 DFT 系数的含义。

4.1.1 DFT 与连续信号频谱关系的直观解释

DFT 的系数 $X[k]$ 也可以直接映射到连续频率,讨论这个问题之前先定义一个新的对